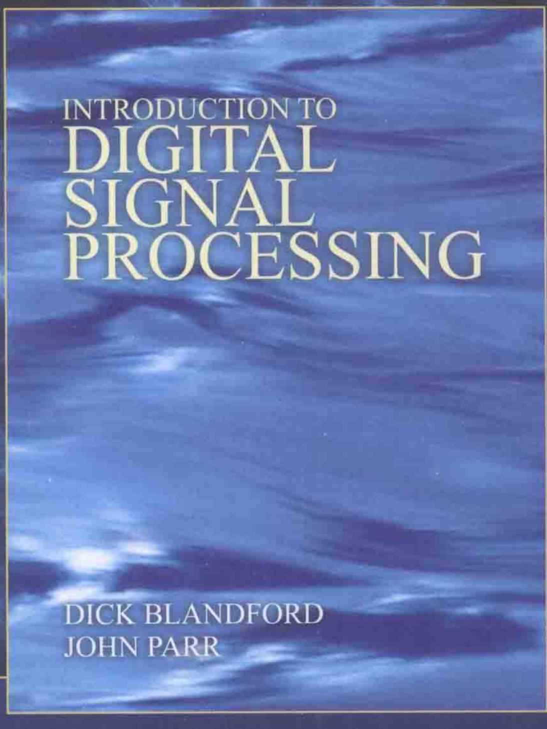
华章教育
PEARSON
国外电子与电气工程技术丛书

数字信号处理及MATLAB仿真

[美] Dick Blandford
John Parr 著
陈后金 李居朋 姚畅 李艳凤 译

Introduction to Digital Signal Processing



机械工业出版社
China Machine Press

本书是美国伊凡斯维尔大学电子与计算机工程专业的DSP课程教材，注重理论与应用相结合，前7章重点讲述数字信号处理基础理论和知识，包括DSP的概述、线性信号和系统概念、频率响应、抽样和重建、数字滤波器的分析和设计、多速率DSP系统；后4章侧重于DSP应用，包括数字滤波器的实现、数字音频系统、二维数字信号处理和小波分析。本书可作为电子信息、通信、控制、仪器仪表等相关专业本科生的DSP课程教材，对初级DSP工程师也是一本实用的参考书。

本书特色

- 内容丰富，实例和案例选材得当，有助于读者理解理论知识，并引导读者深入学习应用内容。
- 前7章涵盖DSP核心内容，既可以按照章节的顺序进行讲解，也可以将第4章放在第5章和第6章的后面讲解，可以让读者对DSP硬件有更多的了解。
- MATLAB仿真贯穿全书。每章后面增加了大量的概念题，有助于读者理解各种变换的计算方法，同时对DSP系统的开发有直接的认识。

译者简介

陈后金 博士，教授，博士生导师，北京交通大学电路与系统博士点学科负责人，教育部电工电子教学指导委员会副主任。主持建设了国家级教学基地、国家级实验教学示范中心、国家级视频公开课程、国家级精品资源共享课程、国家级精品课程、国家级双语教学示范课程、国家级教学团队，主编了多部国家“十五”、“十一五”、“十二五”规划教材和国家级精品教材。近年来，主持20多项国家级教研和科研项目，发表了100多篇研究论文。先后获得了国家级教学成果奖、国家级教学名师奖、宝钢优秀教师特等奖、霍英东青年教师奖，以及多项省部级科技进步奖，被评为铁道部青年科技拔尖人才、教育部新世纪优秀人才、国务院特殊津贴专家、全国优秀教师，入选首批中组部“万人计划”教学名师。

PEARSON
www.pearson.com

投稿热线：(010) 88379604
客服热线：(010) 88378991 88361066
购书热线：(010) 68326294 88379649 68995259

华章网站：www.hzbook.com
网上购书：www.china-pub.com
数字阅读：www.hzmedia.com.cn

封面设计·锡 彬

上架指导：数字信号处理

ISBN 978-7-111-48388-5

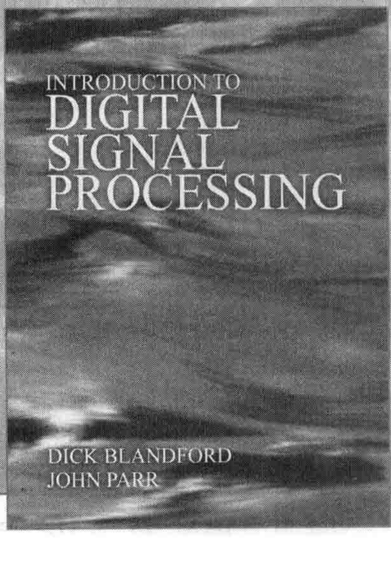
定价：95.00元

国外电子与电气工程技术丛书

数字信号处理及MATLAB仿真

[美] Dick Blandford
John Parr 著
陈后金 李居朋 姚畅 李艳凤 译

Introduction to Digital Signal Processing



机械工业出版社
China Machine Press

图书在版编目（CIP）数据

数字信号处理及 MATLAB 仿真 /（美）布兰福德（Blandford, D.），（美）帕尔（Parr J.）著；陈后金等译．—北京：机械工业出版社，2015.1
（国外电子与电气工程技术丛书）
书名原文：Introduction to Digital Signal Processing

ISBN 978-7-111-48388-5

I. 数…　II. ①布…　②帕…　③陈…　III. 数字信号处理　IV. TN911.72

中国版本图书馆 CIP 数据核字（2014）第 249995 号

本书版权登记号：图字：01-2012-7772

本书讲述基本的数字信号处理知识，主要内容包括数字滤波器、离散时间信号、离散时间系统、频率响应、抽样与重建、FIR 滤波器，IIR 滤波器、抽样速率转换、DSP 的实现、数字音频系统、二维数字信息处理、小波分析等。本书适合作为电子与计算机工程专业学生的中高级课程教材，同时也适合希望自学 DSP 的工程师阅读。

出版发行：机械工业出版社（北京市西城区百万庄大街 22 号　邮政编码：100037）
责任编辑：谢晓芳　　责任校对：董纪丽
印　　刷：北京诚信伟业印刷有限公司　　版　　次：2015 年 1 月第 1 版第 1 次印刷
开　　本：185mm × 260mm　1/16　　印　　张：31
书　　号：ISBN 978-7-111-48388-5　　定　　价：95.00 元

凡购本书，如有缺页、倒页、脱页，由本社发行部调换
客服热线：（010）88378991　88361066　　投稿热线：（010）88379604
购书热线：（010）68326294　88379649　68995259　　读者信箱：hzjsj@hzbook.com

出版者的话

文艺复兴以降，源远流长的科学精神和逐步形成的学术规范，使西方国家在自然科学的各个领域取得了垄断性的优势；也正是这样的传统，使美国在信息技术发展的六十多年间名家辈出、独领风骚。在商业化的进程中，美国的产业界与教育界越来越紧密地结合，信息学科中的许多泰山北斗同时身处科研和教学的最前线，由此而产生的经典科学著作，不仅擘划了研究的范畴，还揭示了学术的源变，既遵循学术规范，又自有学者个性，其价值并不会因年月的流逝而减退。

近年，在全球信息化大潮的推动下，我国的信息产业发展迅猛，对专业人才的需求日益迫切。这对我国教育界和出版界都既是机遇，也是挑战；而专业教材的建设在教育战略上显得举足轻重。在我国信息技术发展时间较短的现状下，美国等发达国家在其信息科学发展的几十年间积淀和发展的经典教材仍有许多值得借鉴之处。因此，引进一批国外优秀教材将对我国教育事业的发展起到积极的推动作用，也是与世界接轨、建设真正的世界一流大学的必由之路。

机械工业出版社华章公司较早意识到“出版要为教育服务”。自 1998 年开始，我们就将工作重点放在了遴选、移译国外优秀教材上。经过多年的不懈努力，我们与 Pearson、McGraw-Hill、Elsevier、John Wiley & Sons、CRC、Springer 等世界著名出版公司建立了良好的合作关系，从他们现有的数百种教材中甄选出 Thomas L. Floyd、Charles K. Alexander、Behzad Razavi、John G. Proakis、Stephen Brown、Allan R. Hambley、Albert Malvino、Mark I. Montrose、David A. Johns 、Peter Wilson 、H. Vincent Poor、Dikshitulu K. Kalluri 、Bhag Singh Guru、Stephane Mallat 等大师名家的经典教材，以“国外电子与电气工程技术丛书”为总称出版，供读者学习、研究及珍藏。这些书籍在读者中树立了良好的口碑，并被许多高校采用为正式教材和参考书籍。其影印版“经典原版书库”作为姊妹篇也越来越多被实施双语教学的学校所采用。

权威的作者、经典的教材、一流的译者、严格的审校、精细的编辑，这些因素使我们的图书有了质量的保证。随着电气与电子信息学科建设的不断完善和教材改革的逐渐深化，教育界对国外电气与电子信息教材的需求和应用都将步入一个新的阶段，我们的目标是尽善尽美，而反馈的意见正是我们达到这一终极目标的重要帮助。华章公司欢迎老师和读者对我们的工作提出建议或给予指正，我们的联系方法如下：

华章网站：www. hzbook. com

电子邮件：hzjsj@hzbook. com

联系电话：(010) 88379604

联系地址：北京市西城区百万庄南街 1 号

邮政编码：100037

HZ BOOKS
华章教育

华章科技图书出版中心

译者序

20 世纪以来，电子信息技术得到了迅速发展，数字化与信息化已深刻影响并改变着我们的生存和发展方式。数字信号处理课程是电子与电气信息类学科专业的核心课程，相关的原理和技术是通信与控制等诸多学科的基础。由于工业化起源于欧美国家，催生了其在电子与电气信息领域有着诸多的发现和发明，特别对信息技术的发展发挥了重要作用。因此，我们需面向世界和未来，引进国外优秀的教学素材，借鉴其先进的教学理念和课程体系，在教学实践中结合国内高等教育的特点，以期提高课程教学效果和人才培养质量。

机械工业出版社引进了系列化的国外教材，本教材由美国 Dick Blandford 和 John Parr 教授主编。该教材体系结构清晰，内容翔实丰富，特别注重理论技术与工程实际的紧密结合。教材中示例和案例选材得当，有助于读者理解相关内容并有助于引导读者尝试应用。在翻译该教材的过程中，充分领会原作者的教学意图，并结合自身多年教学经验对部分文字或内容进行了适当意译或修改（英文原书更像是教学手稿，存在百余处文字错误或表述瑕疵，中文翻译版已全部校正），以达精益求精，期望此教材译本对读者有所裨益，对国内相关教材建设具有积极影响。

全书共 11 章，其中第 1 章、第 10 章和附录由姚畅翻译，第 8 章和第 9 章由李居朋翻译，第 4 章和第 7 章由李艳凤翻译，其余内容由陈后金翻译，并对全书译稿进行了统稿和校核。此外，韩振中、曹霖、陈通、刘睿欣和李冰洁也在书稿翻译中给予了很多帮助，在此一并表示感谢。

陈后金

2014 年 8 月

前　言

历史背景

像许多电子工程课程一样，数字信号处理（Digital Signal Processing，DSP）最初是一门研究生课程，近 30 年来，其逐渐向本科课程渗透，成为电子与计算机工程的本科课程。由于技术的发展，DSP 的应用无处不在，这也是其成为本科课程的主要原因之一。大多数的汽车、几乎每一个家庭、我们随身携带的很多个人物品都包含 DSP 的应用。在一些本科课程中，DSP 通常作为高年级的选修课。在很多教学大纲中，DSP 课程已成为必修课，或者集成到其他必修课中，作为其中的一部分。

概述

本书主要作为电子与计算机工程专业学生的专业课教材，同时也适合作为对 DSP 不太了解的工程师的自学教材。我们写这本书的目的在于，我们认为电子与计算机工程专业的本科生有必要学习 DSP 的导论性课程。在写这本书的过程中，我们假设学生已经掌握了差分方程和线性信号与系统的相关知识。

本书的核心在于介绍 DSP 中的基本概念，这些概念是电子与计算机工程系的每个学生应该理解的。这些核心是以伊凡斯维尔大学大三第二学期学生在该课程学习中遇到的问题为基础的，核心内容主要在前 7 章体现。我们建议以本书为教材的课程包含前 7 章的内容，可选择性地增加一或两章后面章节的内容。

同时，我们建议将本书作为本科生 DSP 相关课程学习的最后一门课程，只有少数学生会继续深入研究这个领域，大部分学生将会研究微电子、计算机、电力系统等。本书侧重于应用，建议在课程中给学生提供相应的实验室，并布置一些与实验相关的作业。

MATLAB 在本书中应用很多，在实际编程中，我们一般使用 C 语言或者其他类似的语言，但在该课的教学中，使用 MATLAB 仿真。在本书中，我们也完成了 C 语言编程的几个例子，这些例子可在 32 位微控制器或专用的 DSP 系统平台上运行。

本书每章后面的习题设置与大多数教材有所不同，除了标准的分析设计题外，我们增加了大量的概念题。分析设计题的答案一般为数值、图或设计，而概念题的答案一般是一些定性的解释。增加概念题的目的在于，我们希望学生不仅能做数学计算，而且对 DSP 系统的开发有直接的认识。

内容安排

前 7 章主要包含电子与计算机工程师应该了解的基本数字信号处理知识，我们希望使用本书的每个人理解这 7 章，同时对于讲授课程的老师来说，建议按照章节的顺序进行讲解。但对于侧重使用 DSP 硬件的老师来说，将第 4 章放在第 5 章和第 6 章的后面也是可行的，这样的好处是能让学生对 DSP 硬件有更多的了解，不足之处是学生在了解抽样过程之前就使用了这个概念。

第 1 章是对 DSP 的简单介绍，包含 DSP 的概述、一些注意事项和一个双线性变换求导的例子，回答了“什么是数字滤波器”的问题，一般在课程的第一天讲解该章。

第 2 章和第 3 章包含基本的线性信号和系统概念，是后面章节的基础。第 2 章介绍离散时间信号和系统的特点与性质，该章节涉及较少新的概念，大部分为介绍性知识，一般用两节课的时间可以讲完。第 3 章介绍频率响应的基本概念，该章由傅里叶级数出发，引出傅里叶变换、离散时间傅里叶变换、离散傅里

叶变换、拉普拉斯变换和 z 变换。对于大多数的学生来说，可以粗略地阅读该章的大部分，但是对 z 变换可能需要引起较多的重视，该章一般需要 4～7 个学时。

第 4 章介绍了抽样和重建，该章包含模-数转换、δ-Δ 转换器、抗混叠滤波器、过抽样、抗镜像滤波器和信号重建，一般需要 3 个学时。对于使用硬件实验室的老师来说，学生可以进行包含抽样理论、混叠、模-数转换和数-模转换的相关实验。

第 5 章和第 6 章介绍数字滤波器的分析和设计，首先第 5 章介绍 FIR 滤波器，IIR 滤波器在第 6 章介绍，重点应放在使用标准方法对 FIR 与 IIR 滤波器进行设计。这两章包含经典的滤波器设计，如窗函数法、帕克斯-麦克莱伦法、脉冲响应不变法、巴特沃斯滤波器、切比雪夫滤波器、双线性变换以及 IIR 滤波器的直接设计，这两章至少需要 20 个学时。对于使用硬件系统的老师来说，学生应该编写相应的程序在所设计的滤波器系统上运行。

第 7 章介绍多速率 DSP，主要包括抽取、内插、抽样速率转换和多速率 DSP 系统的应用，该章节较短，可用 3 个学时完成。

对于对 DSP 要求不高的课程来说，剩下的 5～7 个学时可进行选择性讲解，第 8～11 章为 DSP 系统的应用，为前 7 章的理论知识提供应用的领域，每一章都是独立的，仅和前 7 章的内容相关。第 8～11 章的顺序可以是任意的，一般仅需包含一到两章即可。

第 8 章讨论 DSP 的实现，包含 FIR 和 IIR 实现过程中速度和位数的限制，定点和浮点操作，量化误差和噪声，同时提供一些例题。所有代码是以 C 或类似语言编写的。

第 9 章介绍数字音频系统，该章强调 MATLAB 的非实时应用，主要内容为音频滤波器、过抽样、混响、量化和其他音效。MATLAB 可以读取和写入 wav 文件，可以有效地对该章的例题进行验证。

第 10 章介绍二维数字信号处理，该章节以二维傅里叶变换和二维快速傅里叶变换开始，同时介绍二维离散余弦变换（DCT）。该章仅对傅里叶变换进行了简要的介绍，剩下的大部分用来介绍二维 FIR 滤波器。该章使用 MATLAB 的图像处理工具箱。

第 11 章介绍小波分析，重点介绍 Haar 小波，尽管在实际应用中很少使用 Haar 小波，但其很容易理解。该章还介绍能量紧支撑和多分辨率分析，Daubechies 小波在去噪和信号压缩中的应用。该章的主要目标是让学生对小波的应用有一定的了解。该章还介绍连续小波变换和离散小波变换，使用了 MATLAB 的小波工具箱。

附录介绍本书的一些相关资料。

附录 A 介绍模拟滤波器设计，大多数 IIR 滤波器设计需要依赖模拟滤波器的设计，之后使用映射函数将 s 域映射到 z 域。附录对经典滤波器（巴特沃斯、切比雪夫和椭圆滤波器）的设计进行了概述，之后介绍了拉普拉斯域的传输函数。

附录 B 提供每章主要的参考文献。

附录 C 提供数学背景知识，包括矩阵操作和基本的状态变量方程。

附录 D 列出每一章中使用到的 MATLAB 函数，及其相应的介绍。

本书还提供了部分习题的答案。

致谢

作者首先感谢 James Reising 在本书撰写过程中的支持和鼓励，Jim 对每个章节进行了校对并提出了许多有用的意见，作者对此表示感谢。作者同时感谢在过去十年中学习 EE 311 课程的学生，他们提供了很多独特的见解，这些见解在帮助学生理解离散时间系统和数字信号处理概念有很大帮助。我们还要感谢我们的妻子，Ann 和 Judith，本书的完成与她们的支持和鼓励分不开。

作者对审阅本书内容的审稿人表示感谢：

Ikhlas Abdel-Qader，西密歇根大学（Western Michigan University）

Mahmood Nahvi，加州理工州立大学圣路易斯奥比斯波分校（California Polytechnic State University-San Luis Obispo）

Zhi Ding，加州大学戴维斯分校（University of California-Davis）

David Salvia，宾州州立大学（Penn State University）

Alan Chiu，路易斯安那理工大学（Louisiana Tech University）

Jason Trobaugh，圣路易斯华盛顿大学（Washington University in St. Louis）

Michael Wakin，科罗拉多矿业大学（Colorado School of Mines）

推荐阅读



作者：Abraham Silberschatz 著
中文翻译版：978-7-111-37529-6，99.00元
本科教学版：978-7-111-40085-1，59.00元



作者：Jiawei Han 等著
中文版：978-7-111-39140-1，79.00元



作者：Ian H.Witten 等著
中文版：978-7-111-45381-9，79.00元



作者：Randal E. Bryant 等著
书号：978-7-111-32133-0，99.00元



作者：David A. Patterson John L. Hennessy
中文版：978-7-111-35305-8，99.00元



作者：James F. Kurose 著
书号：978-7-111-45378-9，79.00元



作者：Thomas H. Cormen 等著
书号：978-7-111-40701-0，128.00元

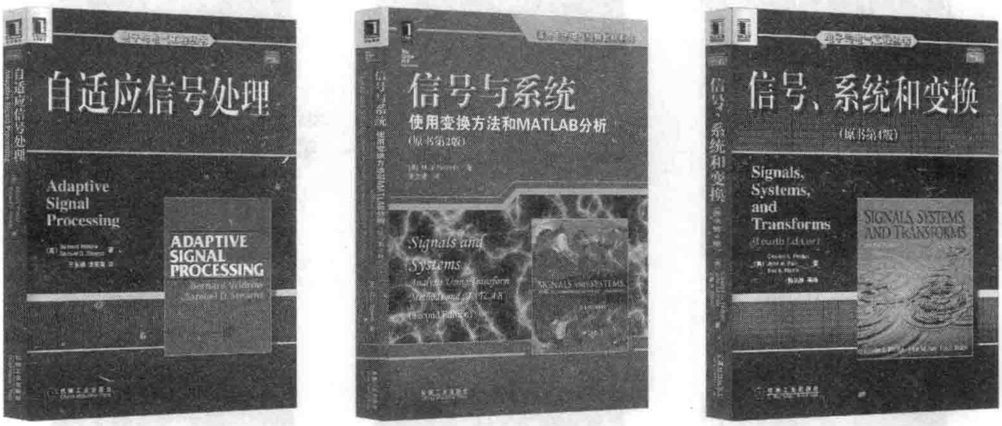


作者：Brian W. Kernighan 等著
书号：978-7-111-12806-0，30.00元



作者：Roger S. Pressman 著
书号：978-7-111-33581-8，79.00元

推荐阅读



自适应信号处理

作者：Bernard Widrow 等 译者：王永德 等 ISBN：978-7-111-22792-2 定价：45.00元

英文版 ISBN：978-7-111-23918-5 定价：56.00元

本本书是一本有关自适应信号处理的专业性教科书，也是自适应信号处理方面的一部经典著作。本书主要内容包括自适应信号处理的原理、算法和结构以及在各种领域的应用。全书在理论方面侧重讨论线性组合器这种基本的自适应结构形式，以及LMS（最小均方）算法这种重要且基本的自适应算法；在应用方面，重点讨论自适应信号处理在系统辨识、解卷积、信道均衡、自动控制、干扰对消及空间阵列处理等方面的应用。

信号与系统：使用变换方法和MATLAB分析（原书第2版）

作者：M. J. Roberts 译者：胡剑凌 预计出版时间：2013年6月

本书在第1版的基础上进行了广泛更新，包含系统的时域分析、傅里叶方法、傅里叶变换、拉普拉斯变换、z变换、采样和信号处理等信号与系统的基本理论和应用。新增加频率响应分析、通信系统分析、滤波器的分析与设计、状态空间分析4章内容。全书给出了大量的例子，并介绍实现上述方法的MATLAB函数和运算。

信号、系统和变换（原书第4版）

作者：Charles L. Philips 等 译者：陈从颜 ISBN：978-7-111-26047-9 定价：78.00元

英文版 ISBN：978-7-111-26894-9 定价：55.00元

本书清晰而全面地叙述了关于信号、系统和变换的理论与应用。介绍了信号与系统的数学背景，主要包括：傅里叶变换、傅里叶级数、拉普拉斯变换、离散时间傅里叶变换与离散傅里叶变换，z变换。本版在课程体系的组织上为读者不同的侧重需求提供了极大的灵活性。MATLAB示例贯穿于书中各章，同时也将MATLAB学生版本的先进特征体现于例题和习题的应用之中。

目录

第1章 概述

过滤器可以定义为这样一种设备，其通过处理输入数据可获得期望的输出数据，且输出数据具有某种共同的特征。日常生活中过滤器无处不在，只是我们没有留意。汽车通常安装有燃油过滤器、空气过滤器以及机油过滤器，它们可以相应地分离出对发动机有害的污染物，并将分离出的污染物滤除。收音机上的调谐盘通过调节过滤器可从不同频率的无线电波中分离出我们感兴趣的频段，提取出对应该频段的原音频信号，从而使我们听到广播，而没有被选中的其他频段则被滤除。我们的身体也有着许多的过滤器，如从血液中分离出代谢废物的肾脏，从食物残渣中吸收营养物质的小肠等。其中大脑是一个更为特殊的过滤器，它对接收到的来自各个感觉器官的庞大信息进行处理，然后做出相应的感知。通常，我们大脑所进行的大部分信息过滤处理不易察觉。例如，我们从手机通讯录中查找联系人可以看作是一种过滤操作，同样，在嘈杂的房间里与人交谈也是大脑对信息的过滤处理。

过滤器涉及的领域十分广泛，本书将仅对电信号中的数据过滤处理进行分析讨论，并将过滤器称为滤波器(filter)。关注信号频域滤波的原因主要有以下三个方面：

1. 特定信息的提取，需要从整个频率信号中分离出某一特定的频率或频段。例如，电视机的频道选择器通过调节滤波器从而提取特定电视台的频段信号。

2. 信号特征的增强，需要滤除给定信号中的噪声或干扰。例如，慢跑者佩戴的心率监测器采用低通滤波器滤除来自身体各个器官感知到的外部环境高频信号干扰，使得心脏跳动信号获得增强。

3. 信号的多角度分析，需要将信号从一种形式变换为另一种形式。基于傅里叶变换的频谱分析仪就是很好的例子，通过变换可以观察到时域信号的频率成分。

不难看出，上述第1项和第2项与滤波器相关，但第3项与信号变换相关。实际上，大多数人喜欢采用更通用的术语——数字信号处理(Digital Signal Processing，DSP)，而将数字滤波器看作是一种特定形式的数字信号处理。本书将采用这种观点。

依据待处理信号的种类，滤波器一般可以分为模拟滤波器和数字滤波器。模拟滤波器因早期无线电的发展而遍布全球，它的发明主要归功于贝尔实验室的乔治·坎贝尔(George Campbell)。模拟滤波器可以像电阻器和电容器一样简单，最显著的特征是其处理的信号在时间和幅度上均是连续的，可表示为 $y=f(t)$，信号在任意时刻 t 都有相应的 y 值定义，并且(理论上) y 可以取到任意实数值。数字滤波器则是对离散时间信号进行处理，此类信号可以表示为 $y=f(nT)$，其中 n 为整数，常数 T 称为抽样间隔。这里，y 只在 T 的整数倍(T、$2T$、$3T$…)时间点取值，在 $nT \sim (n+1)T$ 之间的时刻没有定义值。实际应用中 y 的幅度也是离散的。例如，如果滤波器系数采用8位二进制数表示，那么 y 只能取 $2^8=256$ 个离散值。

离散时间信号处理已经有很久的历史，甚至早于连续时间信号处理。然而，大部分学者将库利(Cooley)和图基(Tukey)在1968年首次提出的快速傅里叶变换(Fast Fourier Transform，FFT)算法作为数字信号处理诞生的标志[⊖]。FFT算法使得计算机在合理的时间内计算出一个复杂信号的频谱成为可能。

⊖ 快速傅里叶变换在3.5节有详细介绍。

今天，DSP 在许多领域都得到了广泛应用，例如雷达和声呐、生物医学信号分析、语音处理、通信应用、图像处理、音乐存储和处理、地震数据处理、数字电视等。在过去的 50 年里，随着高速、低成本计算机和存储设备的出现，DSP 的应用出现了空前的繁荣。

现实世界中大部分信号都是模拟信号，为了获得数字信号，必须首先经过抽样和量化将模拟信号转变成为二进制数。其中量化器就是模/数转换器(A/D)，抽样器是用来捕获某一时刻模拟信号样本的电子开关。连续两次抽样之间的时间间隔 T 称为抽样间隔，抽样间隔的倒数称为抽样频率：$f_s=1/T$。

数字信号处理器接收经过抽样和量化后的二进制串数据，每个抽样信号对应一串二进制数。获得抽样数据之后，数字信号处理器执行一系列的数值运算，通常包括乘法、加法和移位运算。此外，处理器同时存储抽样值，包括当前的抽样值、历史输入和输出的抽样值。运算完成后，处理器通常将处理结果通过数/模转换器(D/A)输出。数/模转换器将二进制数据转换为时间和幅度上离散的信号，经重建滤波器，信号平滑处理为时间和幅度连续的模拟信号。抽样间隔的选择是该处理过程的关键，原因在于所有运算必须在一个抽样间隔内完成才可实现相应的处理。

数字信号处理过程听起来十分简单，但实际上许多问题伴随着处理过程而出现。傅里叶级数、傅里叶变换和 z 变换是了解处理过程的基础，这些基本理论将在本书第 2 章和第 3 章详细论述。模拟信号的抽样和量化将导致信号失真，例如频谱混叠。量化过程产生量化误差，正确设置抽样频率和量化级数是设计的关键，本书第 4 章将对此进行讨论。

数字滤波器可用差分方程表示为

$$y[n]=b_0x[n]+b_1x[n-1]+\cdots+b_Mx[n-M]+a_1y[n-1]+a_2y[n-2]+\cdots+a_Ny[n-N]$$

其中，x 为输入变量，y 为输出变量，N、M、a_i 和 b_i 为常系数。因此，数字滤波器的设计就是确定这些系数取值。当所有系数 a_i 为零时，滤波器称为有限脉冲响应(Finite Impulse Response，FIR)滤波器，设计方法将在第 5 章讨论。对于更一般的情况，当系数 a_i 不全为零时，滤波器称为无限脉冲响应(Infinite Impulse Response，IIR)滤波器，设计方法将在第 6 章进行详细讨论。现代 DSP 系统中，数字信号可由不同的抽样频率获得，多种抽样频率间的转换将在第 7 章讨论。

本书前 7 章主要介绍数字信号处理的基础理论，第 8 章讨论滤波器的实现，如量化误差、溢出和滤波器结构等。实现是将设计转变成实际可用的滤波器的过程。随着新技术的日新月异，滤波器的实现也在不断发展。

本书最后 3 章主要探讨数字信号处理的应用。第 9 章主要介绍 DSP 在语音信号处理中的应用；第 10 章将对二维信号处理应用进行讨论；小波是 DSP 发挥重要作用的一个新领域，第 11 章将对此进行讨论。接下来将通过具体实例说明“什么是数字滤波器”，内容将涉及部分数学推导和直观认知。

1.1　什么是数字滤波器

若要回答这个问题，首先要建立一个简单的模拟滤波器的概念，然后再来说明怎样将模拟滤波器转换成数字滤波器。图 1.1 所示为一个 RC 电路构成的模拟低通滤波器，低频模拟信号可以顺利通过而高频模拟信号的幅度将被衰减。这是因为电路中电容容抗具有随通过的电流频率增加而降低的特性。高频电流通过时，电容容抗很小，近似短路，因而当输入电流的频率上升时，输出电压 v_o 瞬间变得很小。

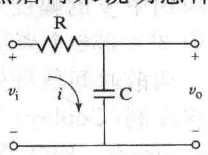


图 1.1　RC 电路构成的低通滤波器

1.1.1　模拟电路的分析

分析图 1.1 中的 RC 模拟低通滤波器电路可以采用多种方法。在时域可通过微分方程

建立输出电压与输入电压之间的关系：

$$v_o = v_c$$
$$v_i = iR + v_c$$
$$i = C\frac{dv_c}{dt} = \frac{Cdv_o}{dt}$$

可得

$$\frac{dv_o}{dt} + \frac{v_o}{RC} = \frac{v_i}{RC} \tag{1.1}$$

要计算出电路的正弦频率响应，首先将输入写成复指数形式

$$v_i = e^{j\omega t} \tag{1.2}$$

利用复指数信号在运算中的数学便利性，式(1.1)可写为

$$\frac{dv_o}{dt} + \frac{v_o}{RC} = \frac{1}{RC}e^{j\omega t}$$

可得微分方程的解为

$$v_o = K \cdot e^{-t/RC} + \frac{e^{j\omega t}}{1 + j\omega RC} \tag{1.3}$$

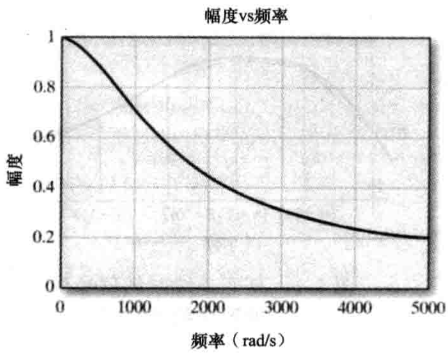
式(1.3)右边第一项为瞬态响应，K 的取值由初始条件决定，第二项为输入正弦信号的稳态响应。由于此为线性系统，因此输出信号的频率将与输入信号的频率相同，但幅度和相位与输入信号不同。

如果我们定义电路的增益为输出与输入之比，那么可画出稳态响应时增益的幅频特性曲线和相频特性曲线。由于输入的幅度恒为1，因此整个电路增益的幅度和相位取决于输出。

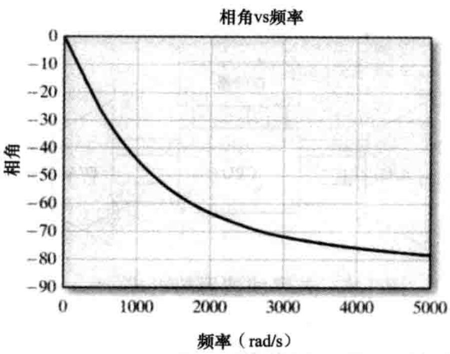
$$增益的幅度 = |H(j\omega)| = \left|\frac{1}{1 + j\omega RC}\right| \tag{1.4}$$

$$增益的相位 = 0 - \angle(1 + j\omega RC) = -\angle(1 + j\omega RC) \tag{1.5}$$

与期望的结果一致，从图1.2可看出滤波器在低频时的增益接近于1，而在高频时的增益趋于0。



a）幅度响应曲线



b）相位响应曲线

图1.2 幅度和相位响应曲线

同样，通过计算微分方程瞬态响应可以获得电路对标准测试信号(如单位阶跃信号)的响应。图1.3所示为电路的阶跃响应。

相似的结果也可通过拉普拉斯变换获得。

1.1.2 数字滤波器的原理

构建替换模拟滤波器的数字滤波器所需的组成部分如图1.4所示。模拟信号从左侧输

入，经过滤波处理后从右侧输出。输入和输出之间的电路部分主要对模拟信号进行抽样并转换成数字信号，然后采用相应的数值算法进行滤波，最后将滤波后的数字信号转化成模拟信号。通过分析此电路可以发现，由于在抽样和数值运算过程中计算机产生误差，滤波后的模拟信号只是图 1.1 中 RC 电路输出结果的近似。

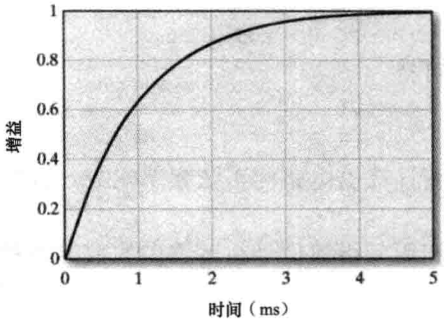


图 1.3　低通 RC 电路的单位阶跃响应曲线，输出表达式为 $v_o=1-e^{-t/RC}$，$t\geqslant 0$

下面通过分析其各个部分来了解数字滤波器的工作原理。输入电压 v_i 通过左侧的抽样器进行抽样。抽样间隔为 T 秒，则抽样频率为

$$f_s=\frac{1}{T}$$

每个抽样值保持 T 秒，直至下一个抽样的到来。在这段时间里，A/D 转换器将模拟信号转换成数字，通常为 10～12 位的二进制数。然后处理器(CPU)采用相应的算法对这些二进制数进行滤波处理。

下面通过两种简单的数字滤波器设计来理解算法如何实现信号的滤波。这两种滤波器的性能与图 1.1 中的模拟滤波器相似。一种设计是将描述 RC 电路的微分方程转换为差分方程，然后利用处理器实现滤波。此种构造的滤波器存在反馈项，称为无限脉冲响应(IIR)数字滤波器。第 6 章将对此进行详细介绍。另一种设计是用 RC 电路的冲激响应来构建与之有相似脉冲响应的离散时间滤波器。这种滤波器没有反馈项，称为有限脉冲响应(FIR)数字滤波器，将在第 5 章详细介绍。

1.1.3　一种简单的 IIR 数字滤波器设计方法

利用数字计算机实现 IIR 数字滤波器，需要将时域连续的微分方程转换为差分方程。目前有多种转换方法，结果各异。其中一种简易的方法是先将微分方程转换为积分方程，然后采用离散求和方法求解积分。图 1.5 所示为一个任意连续时间信号，要计算出该信号的积分，可以通过对图 1.5 中所示的每个小梯形面积求和来近似。

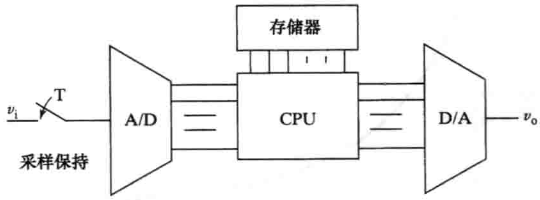


图 1.4　数字滤波器的组成

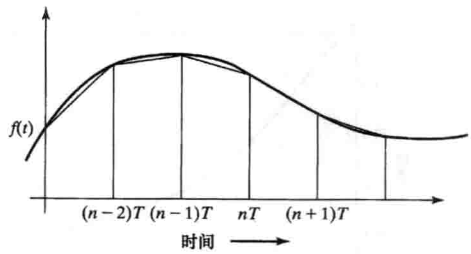


图 1.5　任意连续时间信号积分的梯形面积求和近似

用变量 y 表示积分值，可得

$$y(nT)=\int_0^{nT}f(t)\,\mathrm{d}t \tag{1.6}$$

上式可以分解为两个独立部分的和，即

$$y(nT)=\int_0^{(n-1)T}f(t)\,\mathrm{d}t+\int_{(n-1)T}^{nT}f(t)\,\mathrm{d}t$$

或

$$y(nT)=y([n-1]T)+\int_{(n-1)T}^{nT}f(t)\,\mathrm{d}t \tag{1.7}$$

式(1.7)中，积分可以近似为$(n-1)T\sim nT$之间梯形的面积。该梯形高为T，上底与下底之和为$[f(nT)+f([n-1]T)]$。式(1.7)可写为

$$y(nT)=y([n-1]T)+(T/2)[f(nT)+f([n-1]T)] \tag{1.8}$$

下面将微分方程(1.1)转换为积分方程

$$\frac{\mathrm{d}v_o}{\mathrm{d}t}+\frac{v_o}{RC}=\frac{v_i}{RC}$$

$$v_o+\frac{1}{RC}\int v_o\mathrm{d}t=\frac{1}{RC}\int v_i\cdot\mathrm{d}t$$

或

$$v_o=\frac{1}{RC}\int(v_i-v_o)\cdot\mathrm{d}t \tag{1.9}$$

为得到差分方程，运用式(1.8)的结果做如下代换：

$$v_o(t)\Rightarrow v_o(nT)$$

$$\frac{1}{RC}\int(v_i-v_o)\cdot\mathrm{d}t\Rightarrow v_o([n-1]T)+\frac{T}{2RC}[v_i(nT)-v_o(nT)+v_i([n-1]T)-v_o([n-1]T)]$$

$$\Rightarrow\frac{T}{2RC}[v_i(nT)+v_i([n-1]T)]-\frac{T}{2RC}v_o(nT)+v_o([n-1]T)\left(1-\frac{T}{2RC}\right)$$

因而，式(1.9)可写为

$$v_o(nT)\left(1+\frac{T}{2RC}\right)=\frac{T}{2RC}[v_i(nT)+v_i([n-1]T)]+\left(1-\frac{T}{2RC}\right)v_o([n-1]T$$

整理得

$$v_o(nT)=\frac{T}{2RC+T}v_i(nT)+\frac{T}{2RC+T}v_i([n-1]T)+\frac{2RC-T}{2RC+T}v_o([n-1]T) \tag{1.10}$$

抽样间隔T的大小不是任意的，其取值是DSP系统设计的关键，本书将在后面的章节中详细论述。本例中，取$T=1.0\text{ms}$，$R=1\text{k}$，$C=1\mu\text{F}$，式(1.10)可写为

$$v_o(nT)=0.3333v_i(nT)+0.3333v_i([n-1]T)+0.3333v_o([n-1]T) \tag{1.11}$$

式(1.11)即为得到的差分方程。从该式可以看出，在nT时刻的输出电压v_o可由当前时刻的输入电压、前一抽样时刻的输入电压和输出电压来表示。

差分方程可由图1.4中所示的CPU模块采用如下程序实现：

```
Initialize Variables
DO Forever
    Call A/D(Vi) ;Get a sample from the A/D
    Vo = K1*Vi + K2*Vi1 + K3*Vo1 ;K1 = K2 = K3 = 0.3333
    Call D/A(Vo) ;Output Vo to D/A
    Vo1 = Vo ;Reset the values of the old variables.
    Vi1 = Vi
    Wait for T seconds to pass
Loop
End
```

程序执行时，循环中的所有语句都必须在抽样间隔T秒内执行完毕。

每当一个新的抽样通过A/D转换器后，计算机将对转换后的二进制抽样值进行滤波并将结果发送给D/A转换器。D/A转换器将二进制的滤波结果转换成幅度离散的阶梯状波形信号，每个阶梯时间步长为T秒，如图1.6所示。

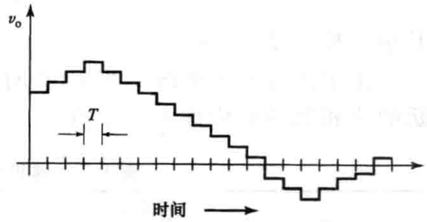


图1.6 D/A转换器输出的典型阶梯状电压信号

由于 D/A 转换器的输出信号在时间和幅度上仍然是离散的，因此在实际应用中，信号还需要通过重构滤波器进行平滑，使得信号在时间和幅度上连续。这部分内容将在第 4 章进行讨论。

若要了解所设计的数字滤波器与 RC 电路对信号滤波结果的近似程度，需要对滤波器进行数学或实证分析。若进行实证分析，可以构造滤波器并输入一个阶跃信号，接着通过观察响应了解什么是滤波器的瞬态响应。也可以输入不同频率的正弦信号，测量输入和输出信号的幅度和相位，然后绘制类似图 1.2 的频率-增益曲线。

数学分析要稍复杂一些。因为阶跃响应最易于计算，所以先考虑该情况。滤波器的差分方程已经由式(1.11)给出。如果输入 v_i 为阶跃信号，即当 $t<0$ 时，$v_i=0$；当 $t\geqslant 0$ 时，$v_i=1$，那么可以计算出输出 v_o 的阶跃响应如表 1.1 所示。

表 1.1 阶跃响应

nT	$v_i(nT)$	$v_o([n-1]T)$	$v_o(nT)$	nT	$v_i(nT)$	$v_o([n-1]T)$	$v_o(nT)$
−1	0	0	0	4	1	0.9752	0.9916
0	1	0	0.3333	5	1	0.9916	0.9971
1	1	0.3333	0.7777	6	1	0.9971	0.9989
2	1	0.7777	0.9258	7	1	0.9989	0.9995
3	1	0.9258	0.9752	8	1	0.9995	0.9997

将表 1.1 中的数据画出数字滤波器的瞬态响应，如图 1.7 所示。由图 1.7 可以看出，数字滤波器的阶跃响应与 RC 电路的连续时间阶跃响应十分接近。

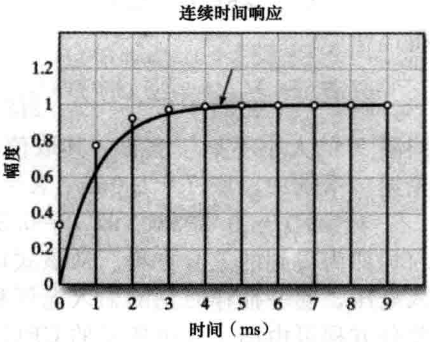


图 1.7 数字滤波器和 RC 电路的阶跃响应，叠加的连续时间阶跃响应有助于比较

例 1-1 已知描述某模拟滤波器的微分方程为

$$\frac{dv_o}{dt}+1000v_o=\frac{dv_i}{dt}$$

抽样频率为 1000Hz，试用梯形面积求和法求出差分方程及其相应的阶跃响应和冲激响应。

解： 先将微分方程转换为积分形式

$$v_o+1000\int v_o dt=v_i$$

利用式(1.8)再转换为离散形式

$$v_i-v_o=1000\int v_o dt$$

$$v_i(nT)-v_o(nT)=v_i([n-1]T)-v_o([n-1]T)+1000T[v_o(nT)+v_o([n-1]T)]/2$$

代入 $T=0.001$ 整理得

$$v_o(nT)=K_1[v_i(nT)-v_i([n-1]T)]+K_2v_o([n-1]T)$$

其中，$K_1=2/3$，$K_2=1/3$。

由于此为因果系统，当 $t<0$ 时所有的输入 v_i 和输出 v_o 的值都为 0，从而可计算出阶跃响应和脉冲响应如表 1.2 所示。◀

表 1.2 通过迭代计算获得的阶跃响应和脉冲响应

	阶跃响应				脉冲响应			
nT	$v_i(nT)$	$v_i([n-1]T)$	$v_o(nT)$	$v_o([n-1]T)$	$v_i(nT)$	$v_i([n-1]T)$	$v_o(nT)$	$v_o([n-1]T)$
−1	0	0	0	0	0	0	0	0
0	1	0	0.667	0	1	0	0.667	0

（续）

	阶跃响应				脉冲响应			
nT	$v_i(nT)$	$v_i([n-1]T)$	$v_o(nT)$	$v_o([n-1]T)$	$v_i(nT)$	$v_i([n-1]T)$	$v_o(nT)$	$v_o([n-1]T)$
1	1	1	0.222	0.667	0	1	−0.444	0.667
2	1	1	0.0741	0.222	0	0	−0.1481	−0.444
3	1	1	0.0247	0.0741	0	0	−0.494	−0.1481
4	1	1	0.0082	0.0247	0	0	−0.165	−0.494
5	1	1	0.0027	0.0082	0	0	−0.0055	−0.165

计算滤波器的正弦响应有些复杂，但同样可以通过给定输入 v_i 的正弦波值，由差分方程迭代计算出输出 v_o。需要注意的是，系统要让正弦波先运行多个周期，因为这里要讨论的是正弦稳态响应，而非起始阶段的瞬态响应。

当输入为抽样后的复指数信号：$v_i(nT)=e^{j\omega nT}$，差分方程(1.11)可写为

$$v_o(nT)=K_1e^{j\omega nT}+K_2e^{j\omega(n-1)T}+K_3v_o([n-1]T)$$

其中，K 为常数，与式(1.11)中的系数一致。

上式还可写为

$$v_o(nT)-K_3v_o([n-1]T)=(K_1+K_2e^{-j\omega T})e^{j\omega nT} \tag{1.12}$$

微分方程可以看作是差分方程的极限情况，因此二者的解在概念上是相似的。式(1.12)既有齐次解(将等号右侧置零后的解)也有特解。齐次解用于求得瞬态响应，并且对于稳定系统，齐次解的幅度随时间而衰减。由激励信号产生的特解可求得系统的稳态响应。

与微分方程的求解类似，式(1.12)的特解可以通过假定输出具有与输入相似的形式来求解。故将 v_o 写为复指数形式

$$v_o(nT)=Ae^{j\omega nT}$$

其中，A 为复数。将 v_o 代入式(1.12)得

$$Ae^{j\omega nT}-K_3Ae^{j\omega(n-1)T}=(K_1+K_2e^{-j\omega T})e^{j\omega nT} \tag{1.13}$$

求得幅度 A 为

$$A=\frac{(K_1+K_2e^{-j\omega T})e^{j\omega nT}}{(1-K_3e^{-j\omega T})e^{j\omega nT}}=\frac{K_1+K_2e^{-j\omega T}}{1-K_3e^{-j\omega T}}$$

因此，复指数激励信号产生的稳态解 v_o 可写为

$$v_o(kT)=\frac{K_1+K_2e^{-j\omega T}}{1-K_3e^{-j\omega T}}e^{j\omega nT} \tag{1.14}$$

由于输入信号是幅度为1的复正弦信号，故滤波器增益可求得

$$\frac{v_o(nT)}{v_i(nT)}=\frac{K_1+K_2e^{-j\omega T}}{1-K_3e^{-j\omega T}} \tag{1.15}$$

该增益为复数，其幅度是一个关于频率的函数。滤波器的幅频特性如图1.8所示，为便于比较，图1.8中同时给出了连续时间RC电路的幅频特性。

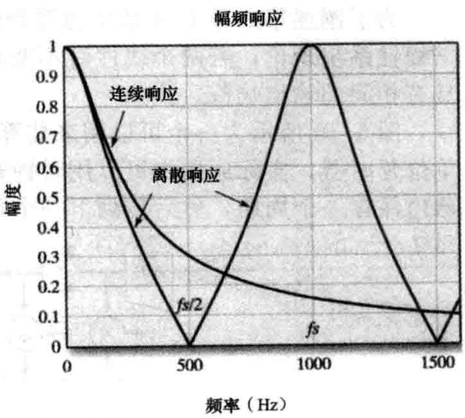


图1.8 数字滤波器和连续时间RC电路的幅频特性比较，抽样频率为1000Hz

由图1.8可以看出，数字滤波器的幅频特性与连续时间滤波器相比有明显差别。但正如第4章(抽样部分)将会讲到，由于混叠现象的原因，所有DSP系统都只有一个有限的可用频率范围。在该数字滤波器中，可用频率范围为0～500Hz，最大频率500Hz正好对应抽样频率的一半。信号的频率若大于抽样频率的一半就会发生混叠，高

于抽样频率一半的频率响应将会对应地混叠到低于抽样频率一半的频率响应。由于信号频率在500～1000Hz之间的响应是0～500Hz之间响应的镜像，故抽样频率的一半常称为折叠频率。任何DSP系统的频率响应都是折叠频率内的响应在整个频率空间上的周期重复。

例 1-2 假设例1-1中差分方程的输入为：$v_i(nT)=e^{j\omega nT}$，画出输出信号的频率响应曲线。抽样频率 $f_s=1000\text{Hz}$。

解：差分方程为

$$v_i(nT)-v_o(nT)=v_i([n-1]T)-v_o([n-1]T)+1000T[v_o(nT)+v_o([n-1]T)]/2$$

令 $v_i(nT)=e^{j\omega nT}$，求得 $v_o(nT)$为

$$v_o(nT)=(2/3)[v_i(nT)-v_i([n-1]T)]+(1/3)v_o([n-1]T)$$

设输出与输入频率相同，即 $v_o(nT)=Ae^{j\omega nT}$，A为复数，可得

$$Ae^{j\omega nT}=(2/3)(e^{j\omega nT}-e^{j\omega(n-1)T})+(1/3)Ae^{j\omega(n-1)T} \tag{1.16}$$

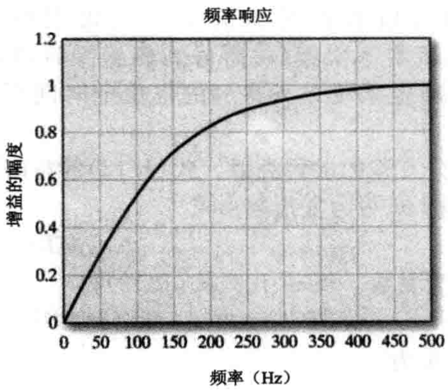
$v_o(nT)$的幅值为$|A|$，解式(1.16)得

$$A=\frac{(2/3)(1-e^{-j\omega T})}{(1-e^{-j\omega T}/3)}$$

利用MATLAB画出$0\sim f_s/2$的频率响应，如图1.9所示。

```
e = exp(1);
fs = 1000;T = 1/fs;
f = 0:fs/2;
w = 2*pi*f;
jwT = j*w*T;
num = abs(2/3*(1 - e.^-jwT));
den = abs(1-e.^-jwT/3);
Amag =num./den;
figure(1);clf;
plot(f, Amag);
axis([0 fs/2 0 1.2]);
xlabel('frequency in Hz');
ylabel('Magnitude of Gain');
title('Frequency plot');
```

a)



b)

图1.9 差分方程的频率响应

从图1.9可知，差分方程描述的是一个高通滤波器。

1.1.4 一种简单的FIR数字滤波器设计方法

为了阐述第二种FIR数字滤波器的设计方法，先从原始RC电路的冲激响应说起。根据线性系统理论，若两个线性时不变系统的冲激响应相同，则它们在相同的输入激励时，具有相同的输出响应。

图1.10所示为一个可以用来估算系统冲激响应的结构图，其中，标有T的模块表示单位延时器，实际应用中可以用时钟寄存器来实现。用二进制表示的离散信号在这些寄存器内保存一个周期，然后传到下一个寄存器。这些排列成一排的寄存器就形成了一条延时

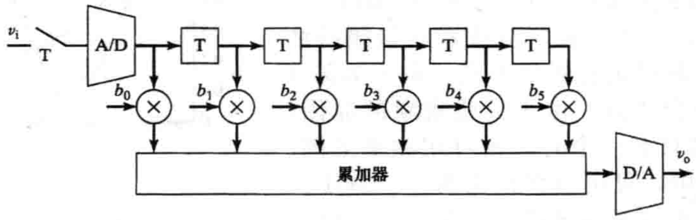


图1.10 有限脉冲响应滤波器

线。在零时刻，给这个系统一个单位脉冲信号，系统响应就可以表示为 $h(kT)=\{b_0, b_1, b_2, b_3, b_4, b_5\}$。用这个方法来构建一个滤波器，我们只需要对冲激响应进行抽样，同时设置 b_i 为抽样的值。

脉冲响应为 $h(kT)=\{b_0, b_1, b_2, b_3, b_4, b_5\}$

图1.1中RC电路的冲激响应为 $h(t)=\frac{1}{RC}e^{-t/RC}$。如果令 $R=1k$，$C=1\mu F$，抽样率为1ms，并且冲激响应的最大值为1，b_i 的抽样值分别为 $b_0=1.0$，$b_1=0.3679$，$b_2=0.1353$，$b_3=0.0498$，$b_4=0.0183$，$b_5=0.0067$，则FIR滤波器的差分方程为

$$v_o(nT)=v_i(nT)+0.3679v_i([n-1]T)+0.1353v_i([n-2]T)+0.0498v_i([n-3]T)+0.0183v_i([n-4]T)+0.0067v_i([n-5]T) \quad (1.17)$$

频率响应函数可以通过将 $v_i(nT)$ 变为复指数形式，求解其等效的差分方程来得到。频率响应幅度曲线如图1.11所示，其中另一条曲线为原RC电路的频率响应曲线。其中只显示了频率响应的一部分，由于抽样的原因，频率响应超出500Hz的部分出现了混叠。

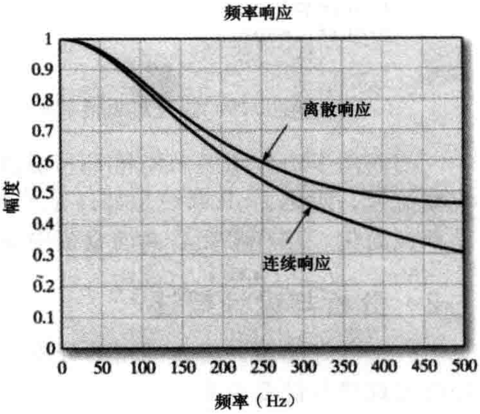


图1.11 6阶FIR滤波器的频率响应与RC低通滤波器的频率响应

例1-3 RLC电路的冲激响应为

$$h(t)=Ke^{-\alpha t}[\omega_d\sin(\omega_d t)-\alpha(1-\cos\omega_d t)]$$

其中，$\alpha=50$，$\omega_d=1200$，$K=0.001$。

对这个冲激响应以频率为1000Hz进行抽样，利用前30个样本构建FIR滤波器。求FIR滤波器的频率响应。

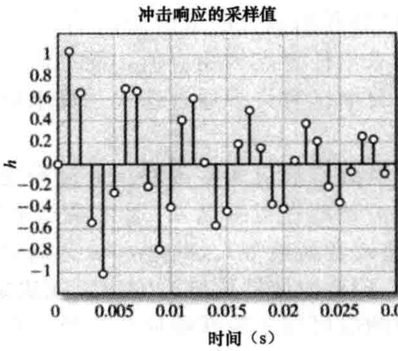
解： 先找到抽样频率为1000Hz时的前30个样本点。差分方程可以写为

$$v_o(nT)=h(0)v_i(nT)+h(1)v_i([n-1]T)+h(2)v_i([n-2]T)+\cdots+h(28)v_i([n-28]T)+h(29)v_i([n-29]T) \quad (1.18)$$

其冲激响应的抽样点图如图1.12所示。

```
fs = 1000;T = 1/fs;
f = 0:fs/2;
Len = 30;
t = 0:T:(Len-1)*T;
a = 50;wd = 1200;K = 0.001;
h = K*exp(-a*t). *(wd*sin(wd*t) - ...
               a*(1 - cos (wd*t)));
figure(1);clf;
stem(t, h);
axis([0 Len*T -2  5.2]);
xlabel('Time in seconds');
ylabel('h');
title('Sampled impulse response');
```

a)



b)

图1.12 $h(t)=Ke^{-\alpha t}[\omega_d\sin(\omega_d t)-\alpha(1-\cos\omega_d t)]$ 的前30个样本

要想找到频率响应，我们令 $v_i(nT)=e^{j\omega nT}$，$v_o(nT)=Ae^{j\omega nT}$，其中，A 为复数。将这两个值带入式(1.18)中，得到

$$Ae^{j\omega nT}=h(0)e^{j\omega nT}+h(1)e^{j\omega(n-1)T}+h(2)e^{j\omega(n-2)T}+\cdots+h(28)e^{j\omega(n-28)T}+h(29)e^{j\omega(n-29)T}$$

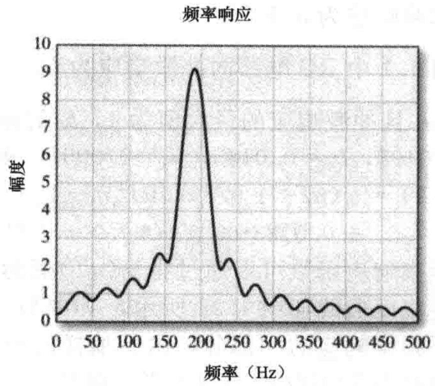
将等式两边同时除以 $e^{j\omega nT}$，得到

$$A = h(0) + h(1)\mathrm{e}^{-\mathrm{j}\omega T} + h(2)\mathrm{e}^{-\mathrm{j}\omega 2T} + \cdots + h(28)\mathrm{e}^{-\mathrm{j}\omega 28T} + h(29)\mathrm{e}^{-\mathrm{j}\omega 29T}$$

利用该等式可以画出 A 的幅频特性曲线。由于输入的幅度为 1，因此 A 的幅度就是电路的增益，图 1.13 所示为滤波器的频率响应。◀

```
w = 2*pi*f;
jwT = j*w*T;
mag = zeros(1, length(f));
for i = 1:Len
    mag = mag + h(i)*exp(jwT*(1-i));
and
mag = abs(mag);
figure(2);clf;
plot(f, mag);
```

a)



b)

图 1.13 滤波器的频率响应图，该图显示电路的共振频率在 200Hz 附近

前面例子中介绍的 IIR 和 FIR 滤波器都与理想滤波器相差较远，这里只是简要介绍其大概流程，对于 DSP 系统是如何工作的提供一个直观认识。在介绍中很多复杂的设计问题都被简化，这些将在后面的章节中深入探讨。

1.2 分析与设计概述

一个良好的设计需要周密的分析以验证其可行，这里将对数字滤波器的分析、设计和最终实现进行简要说明。

1.2.1 分析过程

数字滤波器可从时域或频域进行分析。时域分析通常关注滤波器的瞬态分析，包括脉冲响应和阶跃响应。如果一个系统由差分方程定义，那么总是可以通过输入一个单位脉冲信号或单位阶跃信号，然后采用方程迭代求解的方法获得系统的脉冲响应或阶跃响应。(上一节已经利用该方法求解阶跃响应。)同时，也可以通过闭合形式求解差分方程。闭合形式可以求出任意点的响应值，而不需要首先求解出之前所有点的响应值。为减少运算的单调性，目前已有许多通过迭代运算求解瞬态响应的计算机程序出现，并将结果以图形方式展现。

滤波器瞬态响应非常重要。通过瞬态响应可以判断滤波器的稳定性(相反，频率响应往往不能清楚地进行判断)；通过瞬态响应能够很清晰地判断滤波器的种类(如 FIR 或者 IIR、低通或者高通等)。此外，瞬态响应对 FIR 滤波器而言至关重要，其为设计的核心。(实际上，FIR 滤波器的脉冲响应直接决定了实际应用中滤波器的系数。)

脉冲响应可用于阶跃响应的推导，反之亦然。同样，如果已知完整的脉冲响应，那么可以求解出系统的传输函数(利用 z 变换)和频率响应。

滤波器的频率响应同样非常重要，其是判断滤波器特性是否符合设计指标(通常在频域指定)的测度。依据频率响应曲线形状，滤波器可以分为低通、高通、带通和带阻类型。

采用 z 变换可以很简易地求解滤波器的频率响应。差分方程通过 z 变换由时域变换到 z 域，使得绝大多数数学运算变成了代数运算。代数运算依然冗繁，但可借助于计算机。求解过程与连续域中拉普拉斯变换相似。微分方程通过拉普拉斯变换可由时域变换至 s 域，将 $\mathrm{j}\omega$ 替换 s 将产生一个传输函数，其幅度和相位就是连续信号的频率响应函数。在差

分方程中，将 z 替换为 $e^{j\omega T}$。利用欧拉公式生成关于 ω 的复数方程，即为数字滤波器的频率响应函数。数字系统与模拟系统的求解过程类似，仅数学推导稍许复杂。

1.2.2 设计过程

数字滤波器的设计可分为两个部分——FIR 滤波器设计和 IIR 滤波器设计。FIR 表示有限脉冲响应，该滤波器的特点是不存在反馈项，也就是说，系统差分方程在下一时刻的输出只取决于当前输入值和过去输入值。IIR 表示无限脉冲响应，IIR 滤波器的特点是包含反馈项，这意味着系统差分方程在下一时刻的输出不仅取决于当前输入值和过去输入值，还取决于过去的输出值。

FIR 滤波器可以看作是一个输入信号通过一系列寄存器(数字延迟)的延迟线。其中，每个寄存器接收到输入后将在一个抽样周期内依次将其传输到相邻的寄存器。图 1.10 给出了一个典型 FIR 滤波器的结构。或者，我们可以利用图 1.4 所示的计算机结构通过软件运算差分方程。目前，两种方法均被广泛应用，选择哪一种方法取决于实际应用场合。

图 1.10 中，每个标有 T 的模块表示一个并行输入、并行输出寄存器，每个抽样周期内处于锁存状态。抽样后的输入信号经 A/D 转换器后通过寄存器组，经过 n 个抽样周期后“滑落”到寄存器组的末端。每一个通过寄存器组的抽样信号乘以常数(b_0，b_1，…，b_n)，然后求和并送至 D/A 转换器输出。一个单脉冲信号通过该系统后产生的输出序列为

$$h(nT) = \{b_0, b_1, \cdots, b_n\}$$

滤波器的系数正是系统脉冲响应的数值。由于不存在反馈，因此脉冲响应始终是有限的。

图 1.10 系统结构的差分方程可以写为

$$v_o(nT) = b_0 v_i(nT) + b_1 v_i(n-1)T + b_2 v_i(n-2)T + \cdots + b_m v_m(n-m)T$$

因此，FIR 滤波器的设计在于通过适当方法获得特定频率指标的脉冲响应，其可通过傅里叶反变换实现。由于实际滤波器的脉冲响应需要满足有限和因果要求，因此数字滤波器的频率响应仅是理想滤波器的一个逼近。正是这一逼近往往导致不希望的纹波或其他误差。“加窗”处理可以让设计者去权衡误差产生的位置。一个完整的 FIR 滤波器的设计过程包括频率指标选择、脉冲响应提取，以及利用窗函数修正特定误差。

对比 FIR 滤波器的设计，IIR 滤波器毫无疑问是一个更加开放的设计问题。这将导致更多的权衡和(通常情况下)更复杂的过程。IIR 滤波器设计一般从相应的模拟滤波器设计开始，最后通过将传输函数由 s 域映射到 z 域而完成。有许多方法可以实现 s 域到 z 域的映射，我们将对其中的三种方式进行分析，包括差分法、脉冲响应不变法、双线性变换法。

差分法是最为直接的方法，但当信号频率接近抽样频率的一半时不能获得较好的结果。脉冲响应不变法已经在数字控制系统中获得了广泛应用。双线性变换法是通过模拟滤波器设计实现 IIR 滤波器的使用最广的方法。事实也证明，该方法可以在很宽的频率范围内获得良好的设计效果，在计算机领域获得了大量应用。

由于 IIR 滤波器存在反馈项，使得多种不同结构成为可能。图 1.14 为一个二阶 IIR 滤波器的结构图。

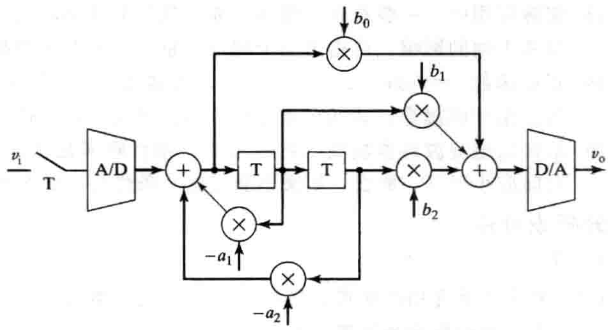


图 1.14 二阶 IIR 滤波器的结构图

对比分析 IIR 和 FIR 数字滤波器，一般而言，对于给定的设计指标，IIR 滤波器往往可以更低的系统阶数满足

指标要求。那么，为什么还需要FIR滤波器呢？事实上，在IIR滤波器的设计中，s域到z域的转换以非线性方式改变了滤波器的相位特征。换言之，IIR滤波器的设计获得了较为准确的幅度响应，但未考虑滤波器的相位响应。在某些应用中，例如，通信领域的相位载波中，这种相位畸变是无法容忍的。在这些情况下，FIR滤波器由于其可以设计成线性相位而获得应用。事实上，FIR滤波器常用在需要保证线性相位特性的场合，而在一些对于相位偏移不重要的场合，则采用IIR滤波器来实现。

习题

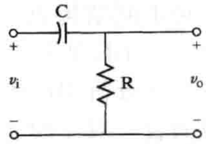
概念题

1. 数字滤波器的稳态响应与瞬态响应有何区别？
2. 请举出三个以上有关线性相位应用的例子。
3. 本章引言部分指出，数字滤波器与模拟滤波器具有相似功能，模拟滤波器由简单的电阻和电容组成，而数字滤波器的硬件实现较为复杂，请问应用数字滤波器替代模拟滤波器有什么优势？从设计灵活性、难易程度、硬件损耗、长期稳定性以及滤波器品质等方面思考。
4. 对于一个给定的算法，为什么人们期望IIR滤波器比FIR滤波器计算更有效？
5. FIR滤波器没有反馈回路，对于系统稳定性来说会产生什么影响？是否会出现震荡现象？并说明原因。
6. IIR滤波器存在反馈回路而FIR滤波器没有。一般来说，两种滤波器系数存在小误差时会带来什么影响？对于IIR滤波器来说，误差的影响是更大还是更小？为什么？
7. 梯形积分法可用于积分的近似计算，以及微分方程转换成差分方程。该方法中若使用诸如抛物线类似的二阶方程近似而不是直线拟合，将会产生什么样的结果？
8. 由于实际中冲激信号不易近似，请描述如何以其他方式间接获得冲激响应。比如说，利用多数信号发生器易于生成的阶跃信号。
9. 计算机求解系统差分方程时，使用二进制方式表示系数，由于计算机位数的限制，导致系数出现小的误差，称为“量化误差”。A/D转换时同样会出现这类误差。请问在求解差分方程时还有可能产生什么误差？
10. 求解差分方程时，输入变量系数与输入变量时间延迟乘以一个常数，那么在频域上将会出现什么结果？
11. 习题1.7中，FIR滤波器的脉冲响应为
    $h(nT)=\{b_0, b_1, b_2, b_3, b_4, b_5\}$
    请画出满足下式的滤波器结构，
    $h(nT)=\{0, 0, 0, b_0, b_1, b_2, b_3, b_4, b_5\}$
    试确定补零操作在滤波器的幅频响应和相频响应上的影响。
12. 假设FIR滤波器差分方程的脉冲响应是对称的，如：
    $h(nT)=\{b_0, b_1, b_2, b_3, b_4, b_5, b_4, b_3, b_2, b_1, b_0\}$
    如何利用对称性形式写出滤波器的差分方程，以减少所需的乘法运算量？
13. 实际应用中，一般希望使用尽可能多的位数来表示差分方程系数以减少其中的量化误差，但同时会带来不利的影响，请指出三个因增加量化位数产生的缺点。
14. 正弦函数$y=A\sin(2\pi ft)$，其中，A为幅度，f为频率。式中未知变量A和f至少需要两个不同时刻的y值才能确定。请问还需要什么条件才可以唯一确定变量A和f的取值？
15. 给定低通滤波器参数截止频率为f_c，抽样频率是f_s，抽样周期$T=1/f_s$，实际应用中差分方程求解时间应小于T。那么，如果不满足这一条件会有什么样的结果？例如，差分方程求解时间为$1.2T$。

分析设计题

1.1 节

1.1　利用本章介绍的梯形法则数值积分方法，给出与图P1.1所示模拟滤波器近似的数字滤波器差分方程。



图P1.1　模拟滤波器

1.2　下面给出的是一个典型的数字滤波器的计算程序：

```
Initialize Variables
DO Forever
   Call AtoD(Vi) ;Get a sample from the A to D
   Vo = K1*Vi + K2*Vo1
   Call DtoA(Vo) ;Output Vo to D to A
   Vo1 = Vo ;Reset the value of the old variable.
   Wait for T seconds to pass
Loop
End
```

请问：

(a) “Wait for T seconds to pass”该命令起什么作用？如果删除会有什么结果？

(b) 在“Initialize Variables”程序部分，有哪些变量需要初始化？

(c) 如果 A/D 需要 10μs 完成转换，每个乘法运行计算机需要 8μs，每个加法运算需要 0.5μs，D/A 转换需要 0.1μs，近似估计该滤波器最大抽样频率是多少。

1.3 对于差分方程

$$v_o(nT) = 0.3333v_i(nT) + 0.3333v_i([n-1]T) + 0.3333v_o([n-1]T)$$

抽样频率是 1kHz。

(a) 求解并绘制脉冲响应曲线；

(b) 求解并绘制阶跃响应曲线；

(c) 当输入正弦与余弦函数，求解输入与输出的频率响应；

(d) 画出频率从 0 变化到 $f_s/2$ 时的幅频图和相频图。

(e) 该滤波器是 FIR 滤波器还是 IIR 滤波器？说明原因。

1.4 本章使用梯形法则离散近似积分运算。

(a) 使用矩形代替梯形，求解相应的差分方程近似积分运算。

(b) 分别画出一个梯形积分器和一个矩形积分器的频率响应。

1.5 详细描述如何使用正弦信号发生器与示波器确定数字滤波器的增益-频率曲线图。

1.6 证明图 P1.6 所示二阶 IIR 滤波器的差分方程为

$$y_k = b_0x_k + b_1x_{k-1} + b_2x_{k-2} - a_1y_{k-1} - a_2y_{k-2}$$

其中，y_k 表示 k 时刻的输出值，x_k 表示 k 时刻的输入值。

1.7 请给出图 P1.7 所示的 FIR 滤波器脉冲响应的一般表达式。证明该响应为有限长，而且长度不超过 $n+1$，其中 n 是尾部系数的下标。

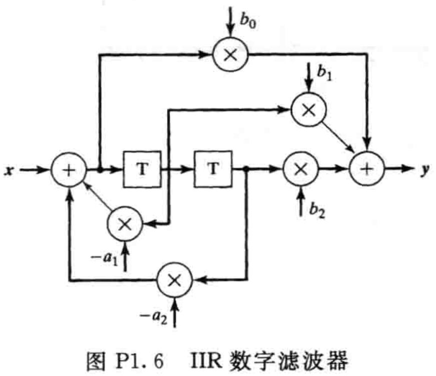


图 P1.6 IIR 数字滤波器

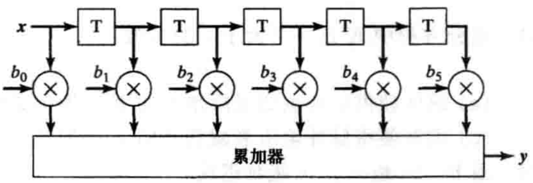


图 P1.7 FIR 数字滤波器

1.8 对于差分方程

$$v_o(kT) = 0.5v_i(kT) - 0.35v_i([k-1]T) - 0.825v_o([k-1]T)$$

(a) 该方程表示的 IIR 滤波器还是 FIR 滤波器？说明原因。

(b) 写出该滤波器脉冲响应的前 5 个点的值。

(c) 抽样频率为 1000Hz，求出系统频率响应的表达式。

1.9 对于差分方程

$$y(nT) = x(nT) + Ky([n-1]T)$$

(a) 取 $T=1$ 时，求出系统频率响应表达式。

(b) 利用 MATLAB 分别绘出 $K=0.8$ 和 $K=1/0.8=1.25$ 时的频率响应曲线，并分析两者的异同点。

(c) 利用 MATLAB 分别绘出 $K=0.8$ 和 $K=1/0.8=1.25$ 时的脉冲响应曲线，至少画出 10 个点。

(d) 对比两个脉冲响应曲线，请总结出频率响应曲线中隐含的规律。

1.10 已知差分方程的脉冲响应为：

$$h(nT)=\{1.0,0.5,0.25,0,0,0,\cdots\}$$

对于线性系统，阶跃输入函数可视为 $nT=0$ 时刻的脉冲序列，阶跃函数的输出应为相应脉冲输入的组合。利用该信息求出系统阶跃响应的前 5 个点的值。

1.11 差分方程的阶跃响应为

$$s(nT)=\{0.25,0.5,1.0,1.0,1.0,\cdots\}$$

阶跃函数在离散时间轴上可以看成是一系列的平移脉冲序列，因此，阶跃响应也应该由一系列的平移脉冲响应构成。利用这个信息，推导出该系统的脉冲响应。

1.12 已知系统差分方程

$$y(nT)=0.075x(nT)+0.075x([n-1]T)+1.5y([n-1]T)-0.65y([n-2]T)$$

系统的频率响应如图 P1.12 所示。抽样频率 $f_s=1000\text{Hz}$。从幅频图和相频图中可以看出输入频率为 100Hz 时，增益约为 0.54，相移约为 $-111°$。利用 MATLAB 生成一个在 100Hz 处幅值为 1.0 的正弦信号 x。应用此差分方程处理 x 并计算输出序列 $y(nT)$。验证增益与相移是否与频率响应图上估计的是否一致。注意，为了得到比较精确的读数，需产生多个周期的 y 值，因此初始的几个周期中不仅包含稳态响应，还包含启动过程中的瞬态响应。

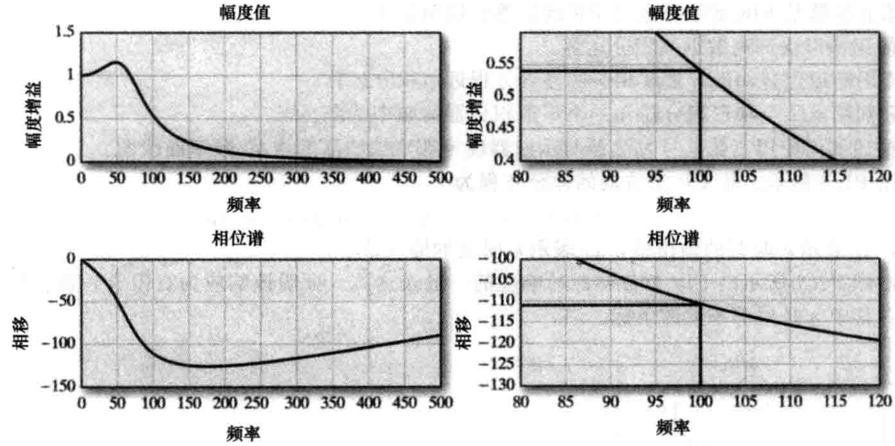


图 P1.12 差分方程的频率响应曲线，右侧为局部放大图

1.13 差分方程中 K 是一个大于 0 的实数，

$$y(nT)=x(nT)+x([n-1]T)+Ky([n-1]T)$$

(a) 求解使得脉冲响应无限增大的 K 的取值范围；

(b) 求解使得脉冲响应衰减到 0 的 K 的取值范围。

1.14 图 P1.14 是一个 N 级延迟线，

(a) 请给出系统差分方程；

(b) 请给出系统频率响应；

(c) 证明频率响应幅度谱总是 1，相移随频率线性变化。

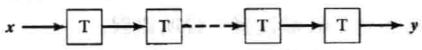


图 P1.14 延迟线结构图

第2章 离散时间信号

本章主要讨论离散时间信号，简称为离散信号。离散时间信号仅仅定义在离散的时刻点上，用 $x(nT)$来表示，也可以简化表示为 $x[n]$。其中自变量 n 只能取整数，而变量 T 表示离散抽样值之间的时间周期。

由于计算机只能对数字进行运算处理，因此连续时间信号必须首先转换成数字序列。这个转换过程称为抽样。如果信号抽样时间间隔是固定的，即产生了序列 $x(nT)$，$n=\cdots$，-2，-1，0，1，2，…，时间间隔 T 称为抽样周期。抽样过程如图 2.1a 所示，其中抽样点的值由垂线的端点表示。

图 2.1b 给出了抽样所使用的常用硬件设备。其中模/数转换器(A/D 或 ADC)是对电压信号抽样并将抽样结果转换成二进制数字的电路，转换之后的二进制数字被传输至计算机进行处理或储存。抽样的时间间隔由计算机的时钟脉冲决定。

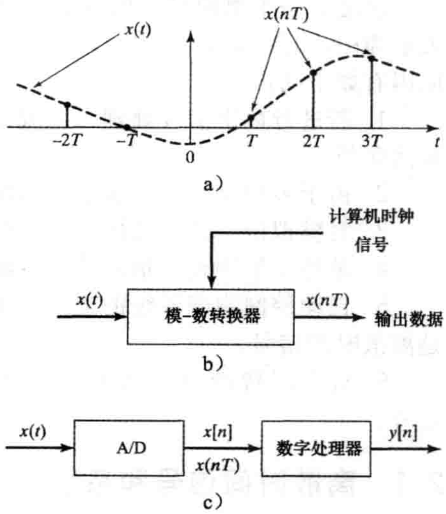


图 2.1　抽样处理硬件流程图

注意信号的表示方法，$x(t)$表示连续时间信号，$x(nT)$表示信号 $x(t)$在特定 $t=nT$ 时刻的值，$x[n]$表示离散时间信号且信号仅在 n 为整数时有定义。另外，书中圆括号表示连续时间，方括号表示离散时间。这种表示方法不是通用的，使用这些表示方法仅仅是为了区分信号 $x[n]$和$x(nT)$。若信号是通过对 $x(t)$进行以 T 秒为周期抽样得到的，那么

$$x(nT) = x(t)\mid_{t=nT}$$

且

$$x[n] = x(t)\mid_{t=nT} = x(nT) \neq x(t)\mid_{t=n} \tag{2.1}$$

图 2.1c 表示了一个完整的数字信号处理系统。抽样器将连续时间信号 $x(t)$转化成离散时间信号 $x(nT)=x[n]$，即完成模拟到数字的转换。由抽样器获得的离散时间抽样值成为数字处理器的输入，数字处理器的输出是 $y[n]$信号。注意，整个系统的输入信号$x(t)$定义在整个时间轴上，而输出信号 $y[n]$仅在时间轴整数点 n 上有定义。

一种离散时间信号的幅度是连续的，其幅度可以取任意值。另一种离散时间信号的幅度是离散的，其幅度只能取特定值。数字信号从本质上来说是幅度离散的离散时间信号。比如模拟/数字转换器的输出 $y[n]$就是一个幅度离散的离散时间信号。假设 A/D 输出的二进制信号用 8 位表示，那么输出信号的幅度只可以取 $2^8=256$ 个不同的值。数字计算机中的任何信号也都是幅度离散的离散时间信号。图 2.2 给出了将-10～10 伏的电压抽样值转换成 3 位二进制数字的结果，其中抽样值用 $2^3=8$ 个电平来表示。因为抽样电压值的范围是 20 伏，所以转换后的二进制信号的取值只能是20/8=2.5 伏的整数倍。图 2.2 中，数据 1 表示抽样值，数据 2 为 3 位二进制数表示的离散幅度序列。

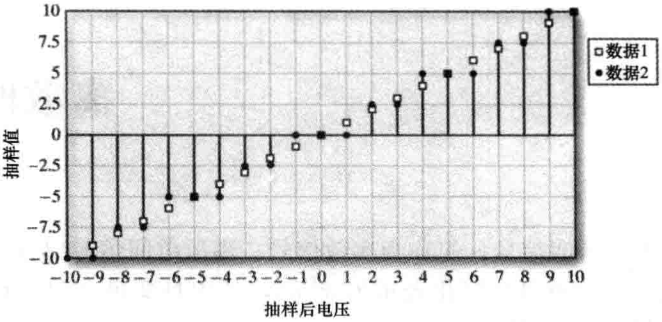


图 2.2　抽样值与二进制转换后数值的比较

为简化本章中一些概念的介绍，通常假设离散时间信号的幅度是连续的，即 $x[n]=x(nT)$。

总之，一个离散时间信号是一个有序的数列。这个数列通常用 $x[n]$表示，整个序列表示为…，$x[-2]$，$x[-1]$，$x[0]$，$x[1]$，$x[2]$，…工程师们对离散时间信号感兴趣的原因有如下几点：

1. 若进行数字信号处理，需要进行信号的抽样，数字信号处理比模拟信号处理更加灵活简便。
2. 由于多种原因，许多通信系统基于离散时间信号的传输。
3. 对模拟信号进行抽样，可以存储其在离散的存储器中。
4. 某些测量物理变量的传感器输出通常都是离散时间信号。
5. 自动控制物理系统的复杂方案需要数字计算机来实现，而计算机输出的控制信号是离散时间信号。
6. 许多消费产品比如电视、CD、DVD、数码相机和 MP3 播放器使用的都是数字信号。

2.1　离散时间信号和系统

本节将通过实例介绍离散时间信号和系统。以数值积分为例，假设我们想利用数字计算机来计算电压信号的积分值。利用计算机进行积分需要使用数值算法，这些数值算法基于将待积分的信号转换成已知积分的近似信号，因此所有的积分算法从本质上来说都体现了近似的概念。

图 2.3 给出了使用欧拉方法进行积分的图示。欧拉方法是将曲线 $x(t)$覆盖的面积近似为所给出的矩形面积之和。其中步长大小 T(矩形宽度)称为数值积分增量，因此实现该算法要求信号$x(t)$必须以 T 秒为周期抽样，得到自变量为 n(整数)的抽样序列。

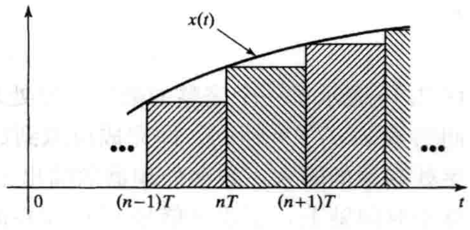


图 2.3　欧拉积分

令 $y(t)$表示如下积分

$$y(t)=\int_0^t x(\tau)\mathrm{d}\tau \tag{2.2}$$

图 2.3 中 $x(t)$从 0 时刻到 nT 时刻的积分就可以表示为 $t=0$ 时刻至 $t=(n-1)T$ 时刻的积分，再加上 $t=(n-1)T$ 时刻至 $t=nT$ 时刻的积分值。因此式(2.2)可以转化为

$$y(t)\Big|_{t=nT}=y(nT)=\int_0^{nT}x(\tau)\mathrm{d}\tau=\int_0^{(n-1)T}x(\tau)\mathrm{d}\tau+\int_{(n-1)T}^{nT}x(\tau)\mathrm{d}\tau \tag{2.3}$$

如果忽略近似误差，使用图 2.3 中的欧拉方法进行积分运算，我们可以将等式改写成

$$y(nT) = y([n-1]T) + Tx([n-1]T) \tag{2.4}$$

然而，通过图 2.3 我们可以看出，$y(nT)$仅仅是对 $x(t)$在 $t=nT$ 时积分的近似估计。

对于离散时间信号来说，式(2.4)可以改写成

$$y[n] = y[n-1] + Tx[n-1] \tag{2.5}$$

此类方程称为差分方程。N 阶线性常系数差分方程一般可表示为如下形式

$$\begin{aligned} y[n] =& a_1 y[n-1] + a_2 y[n-2] + \cdots + a_N y[n-N] \\ &+ b_0 x[n] + b_1 x[n-1] + b_2 x[n-2] + \cdots + b_M x[n-M] \end{aligned} \tag{2.6}$$

方程中系数 a_i 和 b_i，$i=0$，1，2，…，N 为常数，$y[n]$的系数 $a_0=1$。将方程中的变量 n 替换成 $n+N$，则方程可以重新表示为

$$\begin{aligned} y[n+N] =& a_1 y[n+N-1] + a_2 y[n+N-2] + \cdots + a_N y[n+N-N] + b_0 x[n+N] \\ &+ b_1 x[n+N-1] + b_2 x[n+N-2] + \cdots + b_M x[n+N-M] \end{aligned} \tag{2.7}$$

式(2.6)或式(2.7)表示了差分方程的一般形式，差分方程(2.5)～(2.7)定义了离散时间系统。差分方程(2.5)表示的离散时间系统利用欧拉方法计算输入序列 $x[n]$的积分值$y[n]$。第 5 章会继续讨论离散时间系统的差分方程模型，尤其是在数字滤波器中的应用。

通过图 2.3 我们可以看到通过欧拉方法计算的普通函数 $x(t)$的积分可以转化成$x[k]$与常数 T 相乘后的累加和。

$$y[n] = Tx[0] + Tx[1] + Tx[2] + \cdots + Tx[n-1] = T\sum_{k=0}^{n-1} x[k] \tag{2.8}$$

通过这个实例，我们可以看出连续时间上的积分与离散时间上的累加和有一定的联系，而这种联系也广泛适用于其他场合。

2.1.1　单位阶跃函数和单位脉冲函数

我们通过定义两种信号来学习离散时间信号的内容。第一种信号是离散时间单位阶跃函数 $u[n]$，其定义如下

$$u[n] = \begin{cases} 1, & n \geqslant 0 \\ 0, & n < 0 \end{cases} \tag{2.9}$$

这里重新强调这个函数仅对整数 n 有定义。图 2.4a 给出了单位阶跃函数的图示，其中，每条垂直线的端点用来表示离散信号的函数值。有时移的单位阶跃函数表示为 $u[n-n_0]$，其中，n_0 为整数

$$u[n-n_0] = \begin{cases} 1, & n \geqslant n_0 \\ 0, & n < n_0 \end{cases} \tag{2.10}$$

图 2.4b 中给出了当 n_0 为正整数时函数的图形。

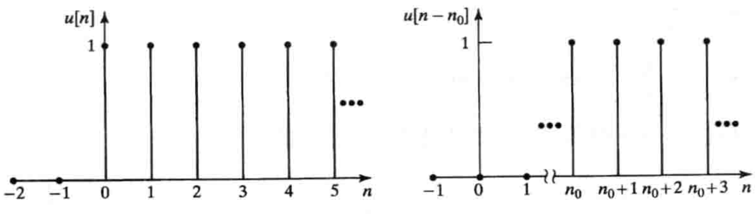


图 2.4　离散时间单位阶跃函数

在 MATLAB 中，一个离散时间单位阶跃函数 $u[n-n_0]$可以通过创建并执行函数 dtstep 来实现。

```
function Y = DTSTEP(X)
% DTSTEP: Discrete-time Step function.
% DTSTEP(X) is 0 for X < 0, 1 for X >= 0.
Y = zeros(size(X));
Y(X >= 0) = 1;
End
```

在函数 dtstep 中，$X=n-n_0$，其中，n 是给定的离散整数向量，n_0 是单位阶跃函数的起始时间点。图 2.4b 可以通过设定 $n_0=2$，按照如下操作重新绘制：

```
>> n = [-1:5];
>> y=dtstep(n-2);
>> stem(n,y)
```

例 2-1 下面的 MATLAB 程序绘制 $u[n-2]-u[n-7]$的图形：

```
>> n=[0:9];
>> y=dtstep(n-2)-dtstep(n-7);
>> stem(n,y)
```

结果如图 2.6 所示。

第二个定义的信号是离散时间单位脉冲函数或单位样点抽样 $\delta[n]$，其定义如下

$$\delta[n]=\begin{cases}1, & n=0\\0, & n\neq 0\end{cases} \tag{2.11}$$

图 2.5 给出了离散时间单位脉冲函数的图形。离散时间单位脉冲函数具有良好的数学特性，其没有任何连续时间冲激函数存在的问题。实际上，离散时间单位脉冲函数可以表示为两个单位阶跃函数之差。◀

$$\delta[n]=u[n]-u[n-1] \tag{2.12}$$

结果可以通过绘制 $u[n]$和 $u[n-1]$的图形得到。时移单位脉冲函数的定义如下

$$\delta[n-n_0]=\begin{cases}1, & n=n_0\\0, & n\neq n_0\end{cases} \tag{2.13}$$

图 2.5b 给出了 $\delta[n-n_0]$在 $n_0>0$ 时的图形。

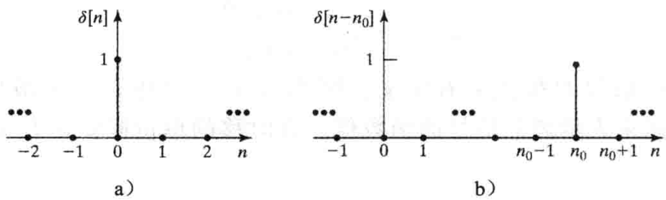


图 2.5 离散时间单位脉冲函数

任意离散时间序列都可以表示为离散时间单位脉冲函数的加权和。离散时间序列由单位脉冲函数表示的一般形式为

$$x[n]=\sum_{m=-\infty}^{\infty}x[m]\delta[n-m] \tag{2.14}$$

例 2-2 利用式(2.14)中的形式来描述图 2.6 中的信号。

$$x[n]=1\delta[n-2]+1\delta[n-3]+1\delta[n-4]1+1\delta[n-5]$$
$$+1\delta[n-6]=\sum_{m=2}^{6}1\delta[n-m]$$ ◀

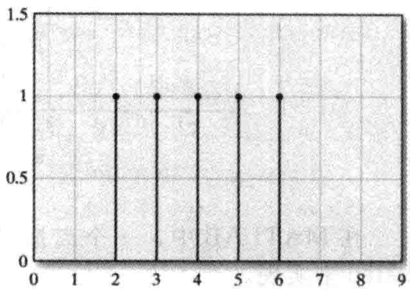


图 2.6 $u[n-2]-u[n-7]$的图形

2.1.2 相关的运算

现在对比一下离散时间信号与连续时间信号的某

些运算。首先，连续时间信号积分对应离散时间信号的累加和。如图 2.7 所示，其中，假定离散时间信号由连续时间信号抽样产生，根据欧拉方法可以得到

$$\int_{-\infty}^{t} x(\tau)\,\mathrm{d}\tau \Leftrightarrow T\sum_{n=-\infty}^{n} x[n] \tag{2.15}$$

其中，$t=nT$。

以类似的方式，通过抽样值 $x[n]$可以近似获得连续时间信号 $x(t)$的斜率，其关系式如下

$$\frac{\mathrm{d}x(t)}{\mathrm{d}t}\Big|_{t=nT} \approx \frac{x[n]-x[n-1]}{T} \tag{2.16}$$

该关系式如图 2.8 所示。式(2.16)右侧分子项称为一阶差分，在某些应用中，也可以看作是连续时间信号一阶导数的等效运算。

$$\frac{\mathrm{d}x(t)}{\mathrm{d}t} \Leftrightarrow x[n]-x[n-1] \tag{2.17}$$

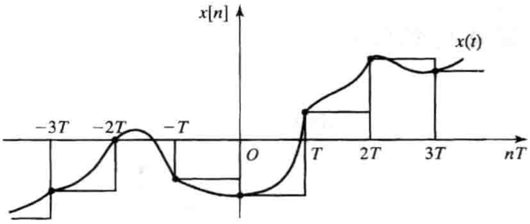


图 2.7　累加运算近似积分运算

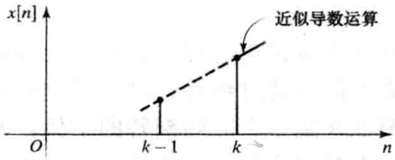


图 2.8　近似微分运算

脉冲函数的等效运算如下

$$x(t)\delta(t) = x(0)\delta(t) \Leftrightarrow x[n]\delta[n] = x[0]\delta[n] \tag{2.18}$$

其中，$\delta(t)$是连续时间或者狄拉克单位冲激。与脉冲函数和阶跃函数相关的运算如下

$$\delta(t) = \frac{\mathrm{d}u(t)}{\mathrm{d}t} \Leftrightarrow \delta[n] = u[n]-u[n-1] \tag{2.19}$$

$$u(t) = \int_{-\infty}^{t} \delta(\tau)\,\mathrm{d}\tau \Leftrightarrow u[n] = \sum_{k=-\infty}^{n} \delta[k] \tag{2.20}$$

作为上式求和的实例，令 $n=3$：

$$u[3] = \sum_{k=-\infty}^{3} \delta[k] = \cdots + \delta[-1] + \delta[0] + \delta[1] + \delta[2] + \delta[3] = \delta[0] = 1$$

因为 $\delta[0]=1$，且当 $n\neq 0$ 时 $\delta[n]=0$。表 2.1 总结了相关的运算。

表 2.1　相关的运算

连续时间	离散时间
1. $\int_{-\infty}^{t} x(\tau)\,\mathrm{d}\tau$	$\sum_{k=-\infty}^{n} x[k]$　(2.15)
2. $\frac{\mathrm{d}x(t)}{\mathrm{d}t}$	$\frac{x[n]-x[n-1]}{T}$　(2.16)，or $x[n]-x[n-1]$　(2.17)
3. $x(t)\delta(t)=x(0)\delta(t)$	$x[n]\delta[n]=x[0]\delta[n]$　(2.18)
4. $\delta(t)=\frac{\mathrm{d}u(t)}{\mathrm{d}t}$	$\delta[n]=u[n]-u[n-1]$　(2.19)
5. $u(t) = \int_{-\infty}^{t} \delta(\tau)\,\mathrm{d}\tau$	$u[n] = \sum_{k=-\infty}^{n} \delta[k]$　(2.20)

2.2 离散时间信号的变换

本节将要介绍离散时间信号 $x[n]$的 6 种变换。其中 3 种变换的对象是自变量 n，另外 3 种变换的对象是变量 $x[\cdot]$。

在命名离散信号变换时，对离散增量 n 会继续使用术语离散时间或时间，因为通常我们考虑的都是抽样后的信号，即对抽样信号，使用 n 来表示时间 $t=nT$，其中，T 是抽样周期。

2.2.1 时间变换

首先讨论三种时间变换(即时间轴变换)，为了使表述更加清晰，令 m 为原始信号中的离散时间变量，n 为变换后信号的离散时间变量。

时间翻转 将信号 $x[m]$的时序进行翻转时，可以将自变量 m 替换为 $-n$，由此可以得到

$$y[n]=x[m]\,|_{m=-n}=x[-n] \tag{2.21}$$

其中，$y[n]$是变换后的信号。时间翻转运算能够得到信号 $x[m]$关于 $m=n=0$ 轴的镜像。

在 2.6 节讨论卷积时，时序翻转将应用于计算输入信号通过线性非时变系统的响应。

图 2.9 是一个时间翻转的实例，其中，原始信号中时间变量由 n 换成 m，根据式(2.21)中 $n=-m$，其中 n 轴绘制在 m 轴下方。由此可以得到变换后 $y[n]$的图形，如图 2.9b 所示。

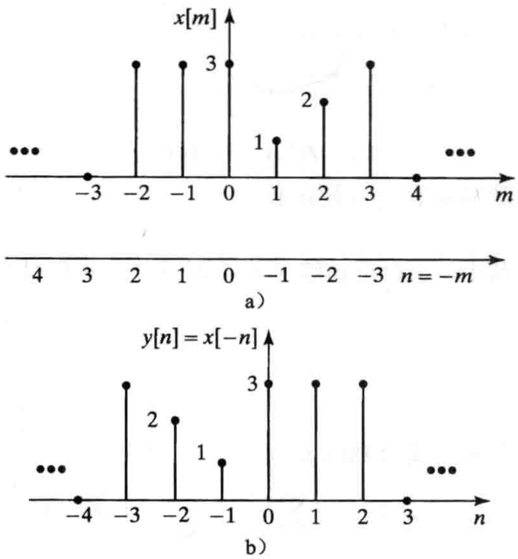


图 2.9 信号时间翻转

时间展缩 对于一个给定信号 $x[n]$，其时间经过扩展或压缩后的信号为

$$y[n]=x[m]\,|_{m=an}=x[an] \tag{2.22}$$

这里只考虑当 $a=k$ 或 $a=1/k$，并且 k 为整数的情况。注意，这里原始信号的时间变量为 m。

图 2.10a 给出了信号 $x[m]$，我们绘制其时间展缩后的图形 $y_1[n]=x[2n]$和 $y_2[n]=x\left[\frac{n}{3}\right]$。对于第一个变换，$m=2n\Rightarrow n=\frac{m}{2}$。

在图 2.10a 中，$\frac{m}{2}$轴显示在 m 轴下方。注意，当 m 为奇数时，$\frac{m}{2}$不是整数，因此当 m 为奇数时，信号 $x[m]$的值并不会显示在 $y_1[n]$中。实际上，在时间缩放后信号 $x[an]$中，当 $a=k\geqslant 2$ 时，信号中部分信息会丢失。如图 2.10a 和 b 所示，m 为奇数时的 $x[m]$在 $x[2n]$中没有出现。

下面根据图 2.10a 所示信号，绘制信号 $y_2[n]=x\left[\frac{n}{3}\right]$的图形。通过变换可得

$$\frac{n}{3}=m\Rightarrow n=3m$$

$3m$ 轴直接显示在图 2.10a 中 m 轴的下方。图 2.10c 给出了 $y_2[n]$的图形。可以看出对于信号 $x[an]$，当 $a=\frac{1}{k}$时，$x\left[\frac{n}{k}\right]$并不是在所有离散增量 n 上都有定义。在图 2.10c 中，

$y_2[n]$在 $n=\pm1$，±2，±4，±5 等时间变量上都没有定义。但是实际中如果要使用信号 $x\left[\frac{n}{3}\right]$，则其必须在所有点上都有定义，因此通常会根据一定的逻辑规则赋予未定义的点函数值，比如内插法。但是如果信号需要实时使用，则应当使用外插法。比如在 $n=0$ 时刻，$y_2[3]$的值未知，因此不能根据内插法得到 $y_2[1]$和 $y_2[2]$的值。

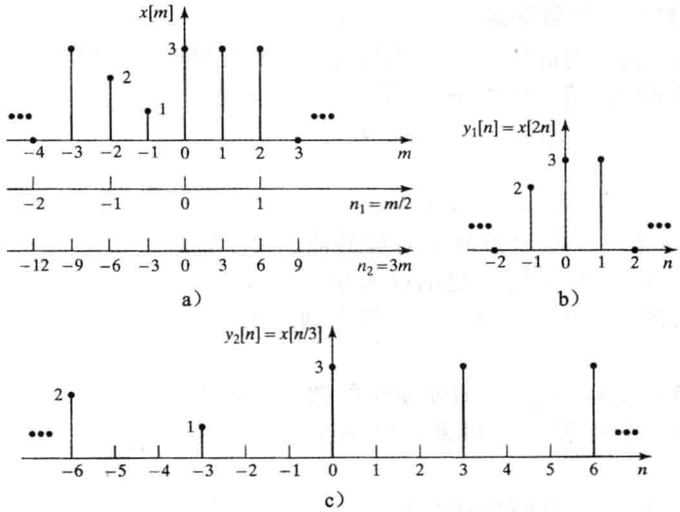


图 2.10 信号时间展缩

时间平移 对于给定信号 $x[m]$，对其进行时间平移后得到信号 $x[n-n_0]$，其中，n_0 为整数常量。

$$y[n]=x[m]\,|_{m=n-n_0}=x[n-n_0] \quad (2.23)$$

2.5 和 2.6 两节将会介绍利用时间平移计算某些系统的响应。

通过实例来介绍时序平移，令

$$x[m]=a^m\cos(\pi m/4)u[m]$$

时移后的信号 $x[n-3]$是 $x[m]$延迟 3 个抽样周期，表示如下

$$\begin{aligned}x[n-3]&=a^{n-3}\cos(\pi(n-3)/4)u[n-3]\\&=a^{n-3}\cos(n\pi/4-3\pi/4)u[n-3]\end{aligned}$$

图 2.11 中给出了 $x[n]$以及 $y_1[n]=x[n-2]$和 $y_2[n]=x[n+1]$的图形。其中，n_1 轴和 n_2 轴由如下变换得到

$$m=n_1-2\Rightarrow n_1=m+2$$
$$m=n_2+1\Rightarrow n_2=m-1$$

图 2.11a 中绘出了 n_1 轴和 n_2 轴，变换后的信号如图 2.11b 和 c 所示。

对于时间平移后的信号 $x[n-n_0]$，$x[0]$的值存在于 $n=n_0$ 点。因此，当 n_0 为正值时，信号的图形向右平移(时间上延迟)；当 n_0 为负值时，信号的图形向左平移(时间上超前)。

本节讨论了三种时间变换，一般形式为

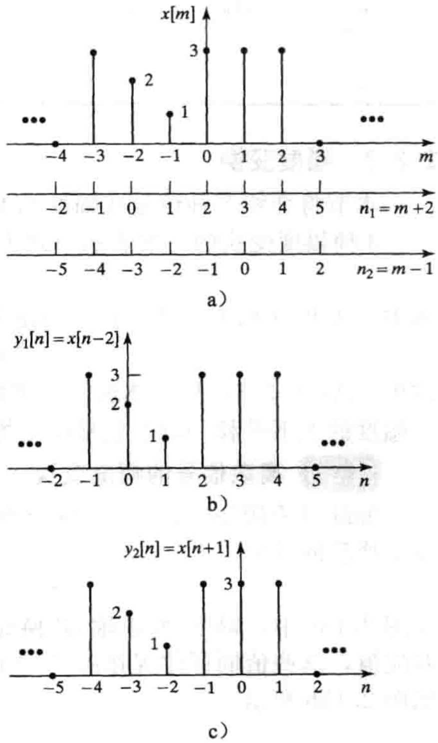


图 2.11 信号时间平移

$$y[n]=x[m]|_{m=an+b}=x[an+b] \tag{2.24}$$

式中，a 是整数或整数的倒数，b 是整数。例如

$$y[n]=x[an+b]=x[-2n+3]$$

其中 $a=-2$，负号体现了时间翻转，$|a|=2$ 体现了时间压缩，$b=3$ 体现了时间平移。下面给出一个时间变换的实例。

例 2-3　离散信号的时间变换

对于图 2.12a 中给出的信号 $x[n]$，我们希望得到经过时间变换 $m=2-n/2$后的信号图形，其中变换包括时间翻转、时间展缩和时间平移。即

$$y[n]=x[m]|_{m=2-n/2}=x[2-n/2]$$

变换表示如下

$$m=2-n/2\Rightarrow n=4-2m$$

表 2.2 给出了当 m 取几个主要点时 n 对应的值。由于仅有 4 个取值点，$y[n]$的对应值也包括在表中。图 2.12a 中，n 轴对应显示在 m 轴下方。表 2.2 的第 3 列给出了 n 轴的取值。

下面绘制变换后的信号 $y[n]$随 n 变化的图形，可以直接根据图 2.12a 或者表 2.2 得到，结果如图 2.12b 所示。

本例展示了两种绘制时间变换后信号图形的方法，可在原始信号图形中加入 n 轴，如图 2.12a 所示，或者可以构造类似于表 2.2 的取值表。◀

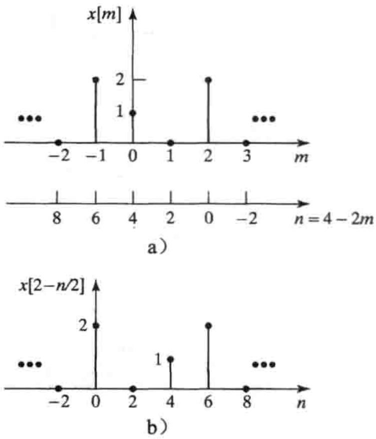


图 2.12　例 2-3 中信号

表 2.2　例 2-3 中的取值

m	$x[m]$	$n=4-2m$	$y[n]$	m	$x[m]$	$n=4-2m$	$y[n]$
−1	2	6	2	1	0	2	0
0	1	4	1	2	2	0	2

2.2.2　幅度变换

本节将介绍 3 种信号在幅度轴上的变换。幅度变换与时间变换有相同的变换规则。

3 种幅度变换的一般表示形式为

$$y[n]=Ax[n]+B \tag{2.25}$$

式中，A 和 B 均为常数，但不一定为整数。比如等式

$$y[n]=-3.5x[n]-6.2$$

式中，$A=-3.5$，负号体现了幅度翻转，$|A|=3.5$ 体现了幅度缩放，而 $B=-6.2$ 体现了幅度的上下平移并改变信号的平均值(直流水平)。下面给出一个信号幅度变换的实例。

例 2-4　离散信号的幅度变换

继续讨论例 2-3 中图 2.12a 所给的信号。图 2.13a 再次给出此信号，我们将绘制其幅度变换后的信号

$$y[n]=3-2x[n]$$

在图 2.13a 中，幅度轴(纵轴)替换成 $3-2x[n]$轴。表 2.3 给出了 $x[n]$取主要点时 $y[n]$的对应值，这些值同样体现在图 2.13a 中的纵轴上。综上可以得到变换后信号 $y[n]$的图形，如图 2.13b 所示。◀

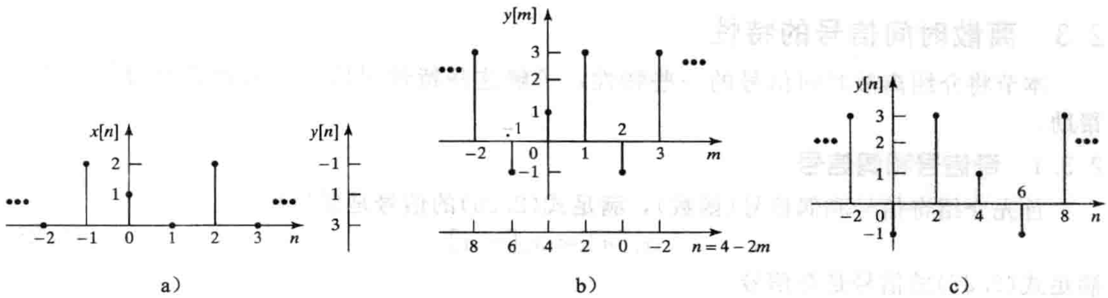


图 2.13　例 2.2 和例 2.3 中的信号

表 2.3　例 2.4 中的取值

$x[n]$	$3-2x[n]$
0	3
1	1
2	−1

例 2-5　离散信号时间和幅度变换

下面讨论信号

$$y[n]=3-2x[2-n/2]$$

其既涉及例 2-3 和图 2.12 中的时间变换，也涉及例 2-4 和图 2.13 中的幅度变换。为绘出该变换后信号的图形，首先利用图 2.13b 中的方法对幅度轴进行变换，然后将图 2.12a 中的 n 轴附加到图 2.13b 中。变换后的信号以 n 为自变量绘制，结果显示在图 2.13c 中。通过改变 n 的值可验证此结果。例如

$$n=0:y[0]=3-2x[2-0]=3-2\times 2=-1$$
$$n=2:y[2]=3-2x[2-1]=3-2\times 0=3$$
$$n=6:y[6]=3-2x[2-3]=3-2\times 2=-1$$

该结果与图 2.13c 相吻合。

下面的 MATLAB 程序实现了例 2-5 中信号的时间和幅度变换。

```
% This MATLAB program performs the time and amplitude
%transformation for Example 2.5.
% Establish vectors of sufficient length for both x(n) and n.
n=[-10:10];
x=zeros(1,length(n));
y=zeros(1,length(n));
k=zeros(1,length(n));
% Enter nonzero values for x(n).
x(10)=2; x(11)=1; x(13)=2;
figure(1), stem(n,x), grid,xlabel('n'),ylabel('x(n)')
for m = 1:length(n);
        k(m)=-2*(m -(length(n)+1)/2)+4;
        y(m)=3-2*x(m);
end
figure(2), stem(k,y), grid,xlabel('n'),ylabel('y(n)')
```

综上所述，离散时间信号的 6 种变换分别是时间的翻转、展缩和平移，以及幅度的翻转、展缩和平移。表 2.4 对这些运算进行了总结。◀

表 2.4　离散时间信号的变换

时间翻转	$x[-n]$
时间展缩	$x[an]$
时间平移	$x[n-n_0]$
幅度翻转	$-x[n]$
幅度展缩	$\lvert A\rvert x[n]$
幅度平移	$x[n]+B$

2.3 离散时间信号的特性

本节将介绍离散时间信号的一些特性，了解这些特性对信号与系统进行分析时很有帮助。

2.3.1 奇信号和偶信号

首先介绍奇信号和偶信号（函数），满足式(2.26)的信号是偶信号

$$x_e[n]=x_e[-n] \tag{2.26}$$

满足式(2.27)的信号是奇信号

$$x_o[-n]=-x_o[-n] \tag{2.27}$$

任意离散时间信号 $x[n]$都可以表示为一个奇信号和一个偶信号的和：

$$x[n]=x_e[n]+x_o[n] \tag{2.28}$$

为了证明这个等式，首先将 n 替换为$-n$，得到

$$x[-n]=x_e[-n]+x_o[-n]=x_e[n]-x_o[-n] \tag{2.29}$$

式(2.28)与式(2.29)相加，可得 $x[n]$的偶信号部分：

$$x_e[n]=\frac{1}{2}(x[n]+x[-n]) \tag{2.30}$$

式(2.28)与式(2.29)相减，可得 $x[n]$的奇信号部分：

$$x_o[n]=\frac{1}{2}(x[n]-x[-n]) \tag{2.31}$$

根据这两个等式，可得到一个离散时间信号的奇信号分量和偶信号分量。式(2.30)和式(2.31)相加，就可得到式(2.28)。

一个离散时间信号的均值定义为

$$A_x=\lim_{N\to\infty}\frac{1}{2N+1}\sum_{k=-N}^{N}x[k] \tag{2.32}$$

离散时间信号的均值取决于其偶信号分量，因为奇信号的均值始终为 0。

奇信号和偶信号具有如下特性：

1. 两个偶信号的和是偶信号。
2. 两个奇信号的和是奇信号。
3. 一个奇信号和一个偶信号的和既不是奇信号也不是偶信号。
4. 两个偶信号的乘积是偶信号。
5. 两个奇信号的乘积是偶信号。
6. 一个奇信号和一个偶信号的乘积是奇信号。

下面给出奇信号和偶信号的实例。

例 2-6 奇函数和偶函数

本例将会得到图 2.14a 中信号 $x[n]$ 的奇分量和偶分量。由于信号只有 6 个非零值，因此可通过简单的数学方法得到结果。利用式(2.30)和式(2.31)，表 2.5给出了结果，表中没有给出的值均为 0。图 2.14 给出了信号 $x[n]$的奇分

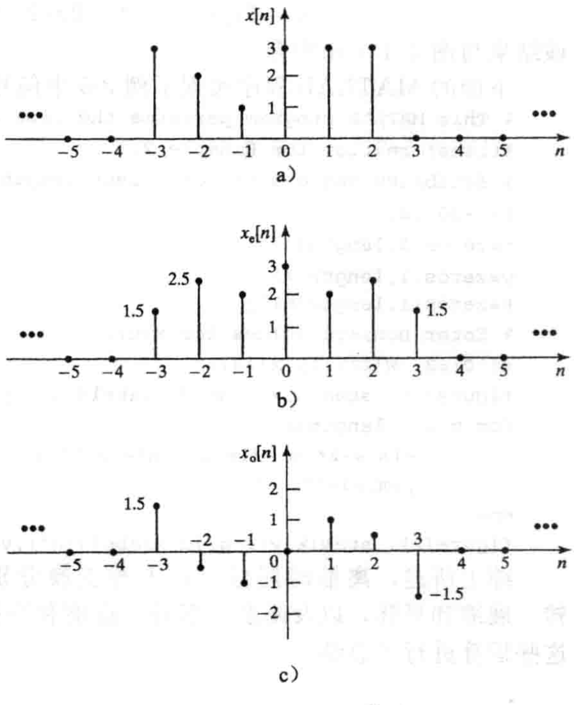


图 2.14 例 2-6 中的信号

量和偶分量的图形。

由表 2.5 可以得到如下三种特性。

1. $x_o[n]$ 信号所有值的和为 0，由式(2.27)可知，n 取任意值时有

$$x_o[n]+x_o[-n]=x_o[n]-x_o[n]=0$$

同时注意，$x_o[0]$的值始终为 0。

2. 累加所有的非零信号值可得到

$$\sum x[n]=\sum x_e[n]$$

3. 对于 n 的每个取值，$x_e[n]$与 $x_o[n]$的和与 $x[n]$相等(根据式(2.28))。

可以通过这三种特性来验证本例的计算结果。

表 2.5　例 2-6 中信号的奇分量和偶分量

n	$x[n]$	$x[-n]$	$x_e[n]$	$x_o[n]$
−3	3	0	1.5	1.5
−2	2	3	2.5	−0.5
−1	1	3	2	−1
0	3	3	3	0
1	3	1	2	1
2	3	2	2.5	0.5
3	0	3	1.5	−1.5

此例的结果可以通过以下 MATLAB 程序实现。◀

```
% Enter the vector of x-values.
x =[0 0 3 2 1 3 3 3 0 0 0];
% Generate the vector of discrete-time increments.
n=[-5:5];
% Reverse x in time.
xr=fliplr(x);
% Apply Equations (2.30) and (2.31).
xe=(x+xr)/2;
xo=(x-xr)/2;
% Plot the original signal,its even part and its odd part.
subplot(3,1,1), stem(n,x)
subplot(3,1,2), stem(n,xe)
subplot(3,1,3), stem(n,xo)
```

2.3.2　以 N 为周期的信号

本节介绍离散时间周期信号。以 N 为周期的离散时间信号 $x[n]$定义如下

$$x[n+N]=x[n] \tag{2.33}$$

其中，n 和 N 都为整数。

首先以 T 为周期对正弦信号 $x(t)=\cos(\omega_0 t)$进行抽样得到信号 $x[n]$，即 $x[n]=x(nT)$。(注意，T 仅仅是抽样周期而并不是信号 $x(t)$的周期。)根据式(2.33)，若 $x[n]$为周期信号，则有

$$x[n]=\cos(n\omega_0 T)=x[n+N]=\cos[(n+N)\omega_0 T]=\cos(n\omega_0 T+N\omega_0 T)$$

已知 $\cos(\theta+2\pi k)=\cos\theta$，所以 $N\omega_0 T$ 必须等于 $2\pi k$(k 为整数)。因此

$$2\pi k=N\omega_0 T=N\frac{2\pi}{T_0}T\Rightarrow\frac{k}{N}=\frac{T}{T_0} \tag{2.34}$$

式中，$T_0=\frac{2\pi}{\omega_0}$是连续时间正弦信号的基本周期。因此抽样周期 T 与正弦信号周期 T_0 的比例必须为整数比，即$\frac{T}{T_0}$必须为有理数。

式(2.34)也可以表示成

$$NT=kT_0 \tag{2.35}$$

该式说明在信号 $\cos(\omega_0 t)$的 k 个周期中必须恰好有 N 个抽样点。这个结论对任何连续时间周期信号的抽样都适用。

定义变量 $\Omega_0=\omega_0 T$，Ω_0 称为归一化离散频率，也可简称为频率。对于抽样后得到的信号，离散频率 Ω 与原始信号频率 ω 通过比例系数(即抽样周期 T)联系在一起。由于 ω 的单位是弧度/秒(rad/s)，而 T 的单位是秒，因此 $\Omega=\omega T$ 的单位是弧度(rad)。

综上所述，若抽样后的信号 $x[n]=\cos(n\omega_0 T)=\cos(n\Omega_0)$ 为周期信号，则必须满足在 k 个周期中恰好有 N 个抽样点。值得注意的是，连续时间周期信号抽样后并不一定能够得到离散时间周期信号。如下例所示。

例 2-7 正弦信号的抽样

本例中将对周期信号 $x(t)=\sin(\pi t)$进行抽样，其周期为 $T_0=\dfrac{2\pi}{\omega_0}=2\text{s}$。首先我们以抽样周期 $T=0.5\text{s}$ 抽样，则在每个正弦信号周期中正好有 4 个抽样点。根据式(2.35)，$kT_0=1\times2=2\text{s}$，$NT=4\times0.5=2\text{s}$。结果如图 2.15a 所示。

下面我们以抽样周期 $T=\dfrac{3}{8}T_0=0.75\text{s}$ 抽样，这样可在每三个周期(也就是 6 秒)中正好有 8 个抽样点。抽样后的信号如图 2.15b 所示。

若以抽样周期为 $T=\dfrac{3}{4}T_0=1.5\text{s}$ 抽样，即在三个周期($4T=3T_0$)中正好得到 4 个抽样点。需要注意的是，虽然在每个正弦信号周期中的抽样点少于两个，但由于满足等式(2.35)，得到的离散时间信号仍然是周期信号。结果如图 2.15c 所示。◀

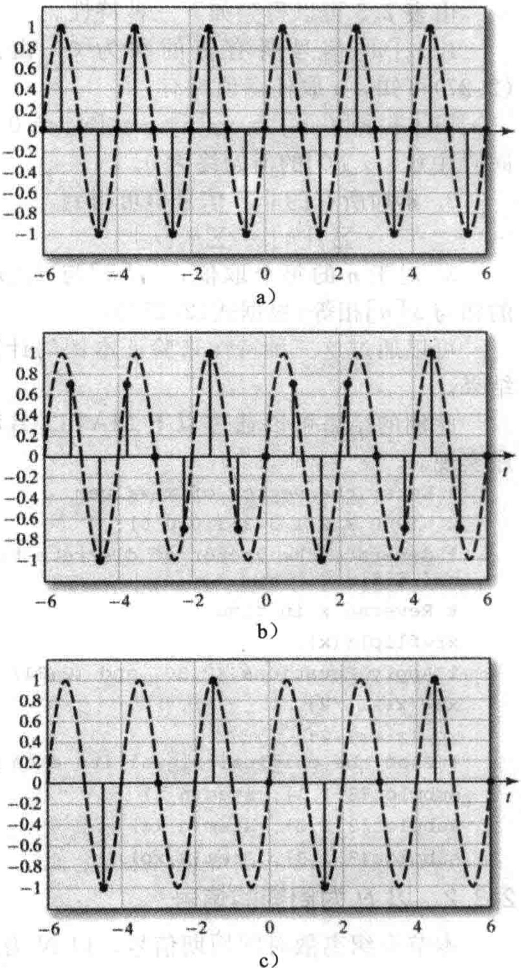


图 2.15　例 2-7 中的周期信号

尽管前面讨论的都是实信号，但上述的结论可直接适用于复指数信号

$$e^{j\omega_0 t}\big|_{t=nT}=e^{jn\Omega_0}=x[n] \tag{2.36}$$

该信号若为周期信号，则存在

$$e^{jn\Omega_0}=x[n]=x[n+N]=e^{j(n+N)\Omega_0}=e^{jn\Omega_0}e^{jN\Omega_0} \tag{2.37}$$

若为周期信号，需要满足 $e^{jN\Omega_0}=e^{j2\pi k}=1$，可以看到 $N\Omega_0=2\pi k$这个条件与式(2.34)相同。

接下来讨论任意的复指数信号，其并非来自连续时间信号的抽样，其表示如下

$$x[n]=e^{jn\Omega_0}=1\angle n\Omega_0 \tag{2.38}$$

该信号可以表示为复平面上角度为 $n\Omega_0$ 的单位向量，如图 2.16所示。这个单位向量在实轴上的投影为 $\cos(n\Omega_0)$，在虚轴上的投影为 $\sin(n\Omega_0)$。信号可以表示为

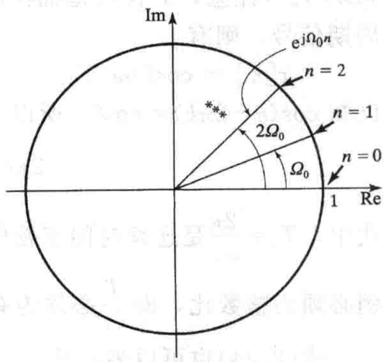


图 2.16　复指数信号的图示

$$e^{jn\Omega_0}=\cos(n\Omega_0)+j\sin(n\Omega_0)$$

我们已经由式(2.36)定义过复指数信号，上述的结论都可以直接应用。对于抽样后得到的

信号，离散频率 Ω_0 与原始信号频率 ω 通过比例系数(即抽样周期 T)联系在一起，变量 $\Omega_0=\omega_0 T$ 的单位是弧度。

现在从另一个方式解释式(2.38)。若式(2.38)定义的信号为周期信号，则有

$$x[n]=e^{jn\Omega_0}=x[n+N]=e^{j(n\Omega_0+N\Omega_0)}=e^{j(n\Omega_0+2\pi k)} \tag{2.39}$$

其中 k 是整数，因此周期需满足

$$N\Omega_0=2\pi k\Rightarrow\Omega_0=\frac{k}{N}2\pi \tag{2.40}$$

即 Ω_0 可表示为 2π 与一个有理数相乘。比如 $x[n]=\cos(2n)$ 不是周期信号，因为 $\Omega_0=2$；$x[n]=\cos(0.1\pi n)$ 则是周期信号，因为 $\Omega_0=0.1\pi$，这时根据式(2.40)得到 $k=1$ 且 $N=20$。

最后说明一点，对于以 N 为周期的复指数周期信号 $e^{jn\Omega_0}$ 需要满足每个周期有 N 个抽样值，那么根据式(2.40)可以得出

$$N=\frac{2\pi k}{\Omega_0} \tag{2.41}$$

在此方程中，k 取能够满足此等式的最小正整数，以保证 N 是一个大于 1 的正整数。比如对于 $x[n]=\cos(0.1\pi n)$，N 的值为

$$N=\frac{2\pi k}{0.1\pi}=20k=20$$
$$k=1 \tag{2.42}$$

如图 2.16 所示绘出该信号，每个周期存在 20 个单位向量($N=20$)。

对于信号 $x[n]=\cos(5\pi n)$，每个周期的抽样点为

$$N=\frac{2\pi k}{5\pi}=0.4k=2,k=5$$

当 $k<5$ 时，N 为非整数。在图 2.16 中绘出该信号，满足每 5 个周期($k=5$)存在两个($N=2$)单位向量。从 $n=0$ 开始的两个向量为 $1\angle 0$ 和 $1\angle 5\pi$，接下来的向量 $1\angle 10\pi=1\angle 0$，即重复了第一个向量。

最后一个例子，令 $x[n]=\cos(2\pi n)$，则

$$N=\frac{2\pi k}{2\pi}=k \tag{2.43}$$

当 $N=k=1$ 时等式成立，因此信号可以表示为

$$x[n]=\cos(2\pi n)=1$$

该离散时间信号为常数。

2.3.3 以 Ω 为周期的信号

前面介绍了复指数信号 $e^{jn\Omega_0}$ 以 n 为周期的条件。实际上，该信号在离散频率变量 Ω 上始终为周期信号。若把信号中变量 Ω_0 替换成($\Omega_0+2\pi$)，即

$$e^{j(\Omega_0+2\pi)n}=e^{jn\Omega_0}e^{j2\pi n}=e^{jn\Omega_0} \tag{2.44}$$

(因为 $e^{j2\pi n}=1$。)可以看到信号 $e^{jn\Omega_0}$ 总是以自变量 $\Omega_0=2\pi$ 为周期，当然，正弦型信号 $\cos(n\Omega_0+\theta)$ 也是以 2π 为周期的信号。此特性在第 4 章讨论信号的抽样时非常重要。

需要注意的是，连续时间周期信号在频域并非周期信号。例如，对于复指数信号

$$e^{j(\omega+a)t}=e^{j\omega t}e^{jat}\neq e^{j\omega t},a\neq 0$$

综上所述，离散时间正弦信号具有两个特性，该特性连续时间正弦信号却不具有。具体阐述如下：连续时间正弦信号 $\cos(n\omega+\theta)$ 始终是周期信号；当且仅当 $\omega_1=\omega_2$ 时，信号 $\cos(\omega_1 t)$ 与 $\cos(\omega_2 t)$ 相同。离散时间正弦信号 $\cos(n\Omega_0+\theta)$ 有如下特性：

1. 根据式(2.40)，若以时间 n 为自变量，信号 $\cos(n\Omega_0+\theta)$ 以 N 为周期，当且仅当

$$\frac{\Omega_0}{2\pi}=\frac{k}{N}$$

其中，k 和 N 为整数

2. $\cos(n\Omega_0)$在频域变量 Ω 上总是以 $2\pi k$ 为周期，k 取任意整数。

$$\cos(n\Omega_0+2\pi k)=\cos(n\Omega_0)$$

这些特性对于 $\cos(n\Omega_0+\theta)$和 $e^{jn\Omega_0}$ 同样适用。

本节介绍了离散时间信号的奇偶性，并对其周期性进行了研究。另外，离散时间信号在频域为周期信号，这个特性在第 4 章介绍连续时间信号的抽样时非常重要。

2.4 常见的离散时间信号

本节将介绍几种常见的离散时间信号，这些信号会出现在离散时间系统的瞬态响应中。

2.3 节已提到正弦信号。例如，利用计算机输出离散时间正弦信号来生成不同频率的音频。该离散正弦信号由计算机传递至数模转换器中，然后通过功率放大器输送电压信号到扬声器。该过程的流程图如图 2.17 所示，其中，计算机的时钟芯片决定了信号的抽样周期，也就决定了声调的频率。

下面通过具体实例来介绍一种常见的离散时间信号。图 2.18a 中的模块是一个存储数据的存储装置。这个存储装置广泛应用于并行输入/并行输出寄存器以及计算机的存储器中。其工作方式为每 T 秒读取存储器中的数据，同时将新的数据锁存其中。假如将锁存到存储器中的数据定义为 $x[n]$，则刚读取的数据即为 $x[n-1]$。这种工作方式叫做理想时延。“理想”指数据仅因时延而非其他任何方式改变。

假设将并行输入/并行输出寄存器连接到系统中，如图 2.18b 所示，标有字母“T”的模块代表寄存器。寄存器读出的值与常数 a 相乘，并作为下一次存入寄存器的数值，则系统可以描述为

$$x[n]=ax[n-1] \tag{2.45}$$

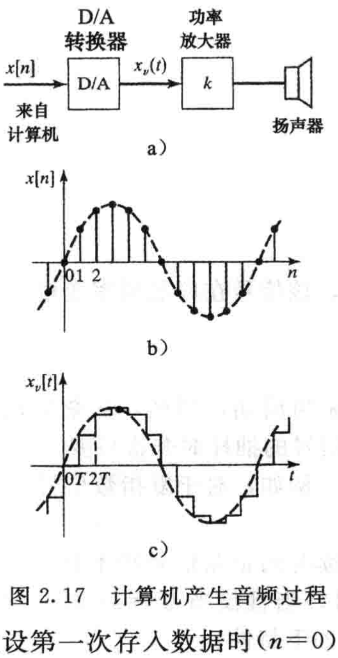


图 2.17 计算机产生音频过程

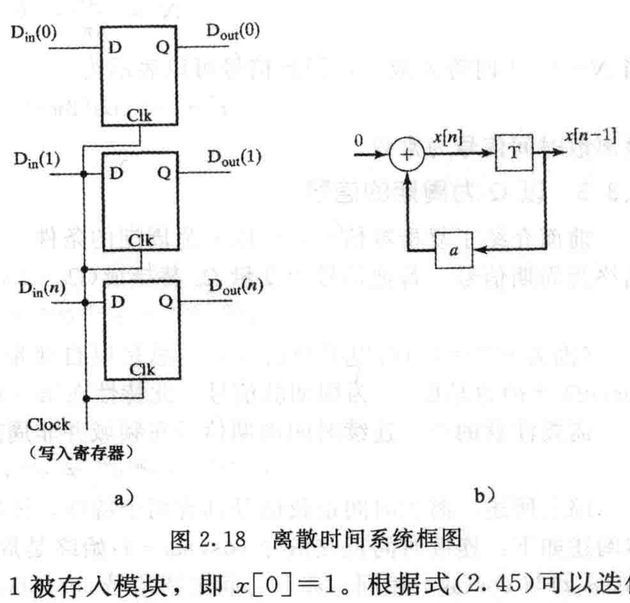


图 2.18 离散时间系统框图

假设第一次存入数据时($n=0$)，数字 1 被存入模块，即 $x[0]=1$。根据式(2.45)可以迭代计算 $x[n]$。

$$x[1] = ax[0] = a$$
$$x[2] = ax[1] = a^2$$
$$x[3] = ax[2] = a^3$$
$$\vdots$$
$$x[n] = ax[n-1] = a^n$$

因此当初始条件 $x[0]=1$ 时，系统输出信号为 $x[n]=a^n$。

当 $a=0.9$ 时，利用 MATLAB 程序对系统进行仿真。

```
a=0.9;
x(1)=1;
for m=2:5
      x(m)=a*x(m-1);
end
x
result: x = [1 0.9 0.81 0.729 0.6561]
```

值得注意的是，在 MATLAB 中要求数组的下标为正整数，因此该程序只能对于 $n=1, 2, 3, 4$ 或 $m=2, 3, 4, 5$ 计算 $(0.9)^n$ 的值。

进一步讨论离散信号 $x[n]=a^n$ 的一些特性。该信号可以表示为离散时间指数函数，假如令 $a=e^b$，则

$$x[n] = a^n = (e^b)^n = e^{bn} \tag{2.46}$$

例如，对于信号 $x[n]=0.9^n$，可以求解此方程得到

$$0.9 = e^b \Rightarrow b = \ln(0.9) = -0.105$$

即

$$x[n] = 0.9^n = e^{-0.105n}$$

通常把离散信号 a^n 看作是离散时间指数信号。

离散信号经常由连续信号抽样得到，假设对指数信号 $x(t)=e^{-\sigma t}$ 进行间隔为 $t=nT$ 的抽样，对于 $\sigma>0$，可得：

$$x[n] = e^{-\sigma nT} = (e^{-\sigma T})^n = a^n \tag{2.47}$$

根据式(2.26)，连续时间指数信号存在时间常数 τ，使得

$$e^{-\sigma t} = e^{-t/\tau} \Rightarrow \sigma = \frac{1}{\tau} \tag{2.48}$$

根据式(2.47)得到

$$x[n] = e^{-nT/\tau} = (e^{-T/\tau})^n = a^n \tag{2.49}$$

通常来说，比例系数 τ/T(即每个时间常数 τ 内的抽样点数)一般不是整数，由式(2.49)

$$e^{-T/\tau} = a \Rightarrow \frac{\tau}{T} = \frac{-1}{\ln a} \tag{2.50}$$

因此可将时间常数 τ 定义为

$$\tau = \frac{-T}{\ln a}$$

该时间常数是根据抽样得到的离散指数信号 a^n 定义的。

例 2-8 离散指数信号的时间常数

根据式(2.50)，离散时间信号 $x[n]=(0.8)^n$ 的时间常数为

$$\frac{\tau}{T} = \frac{-1}{\ln 0.8} = 4.48 \Rightarrow \tau = 4.48T$$

因此，每个时间常数内有 4.48 个抽样值。经过 4 个时间常数后指数函数幅度衰减到可忽略

$$nT > 4\tau = 4(4.48T) = 18T$$

即 $n>18$ 时信号可以被忽略。◀

离散指数信号的一般形式为

$$x[n] = Ca^n \tag{2.51}$$

其中，参数 C 和 a 均可为复数，尽管自然界中不存在复信号。然而，如求解微分的情形一样，在求解差分方程时，假设复信号存在于激励和响应中，会大大简化求解难度。实际工程中只是用复函数的实部或者虚部。下面针对三种情况对复指数信号进行讨论。

情况 1：C 和 a 均为实数

对于第一种情况，C 和 a 为实数。图 2.19 给出了当 C 和 a 为正实数时信号的图形。当 $a>1$ 时，信号随着 n 的增大呈指数型增长；当 $0<a<1$ 时，随着信号随着 n 的增大呈指数型衰减；当 $a=1$ 时，$x[n]=C(1)^n=C$，信号为常数。

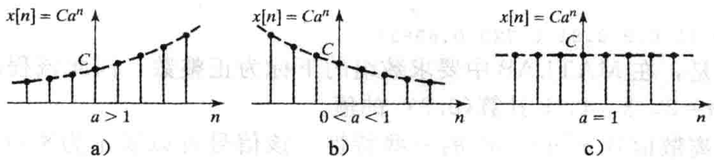


图 2.19　离散时间指数信号

图 2.20 给出了当 $a<0$ 时的三种情况。比如 $x[n]=(-2)^n$，以 $n=0$ 为起始点，则 $x[n]$的序列为 1，−2，4，−8，16，−32，…因此序列呈现正负交替的指数型增长。令 $a=-\alpha$，其中 α 为正值，那么

$$x[n] = Ca^n = C(-\alpha)^n = C(-1)^n\alpha^n = (-1)^n x_a[n] \tag{2.52}$$

其中，$x_a[n]$为图 2.19 中的指数信号。可以看到图 2.20 中信号与图 2.19 中信号有相同的模值，但正负值交替出现。

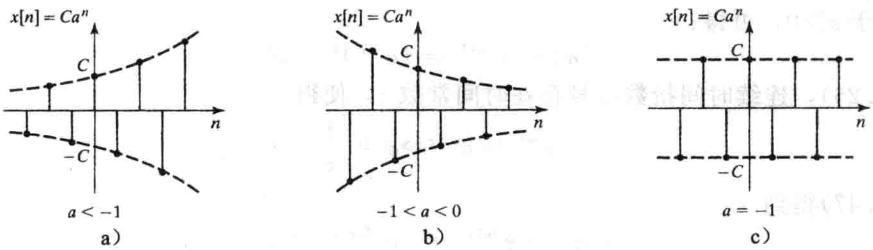


图 2.20　离散时间指数信号

情况 2：C 为复数，a 为复数且模值为 1

接下来讨论 C 和 a 均为复数的情况。

$$C = Ae^{j\phi} = A\angle\phi$$
$$a = e^{j\Omega_0} \tag{2.53}$$

其中，A、ϕ 和 $\Omega_0=\omega_0 T$ 均为实数常量。则根据欧拉关系式复指数信号 $x[n]$可表示为

$$x[n] = Ae^{j\phi}e^{jn\Omega_0} = Ae^{j(n\Omega_0+\phi)} = A\cos(n\Omega_0+\phi)+j\sin(n\Omega_0+\phi) \tag{2.54}$$

在 2.3 节中介绍式(2.54)中的正弦信号，当且仅当 $\Omega_0=2\pi\frac{k}{N}$，并且 k 和 N 均为整数时，其为周期信号，图 2.21 给出了当 $\phi=0$ 时式(2.54)所得信号的实部图形。

情况 3：C 和 a 均为复数

在这种情况下复指数信号 $x[n]=Ca^n$ 的参数

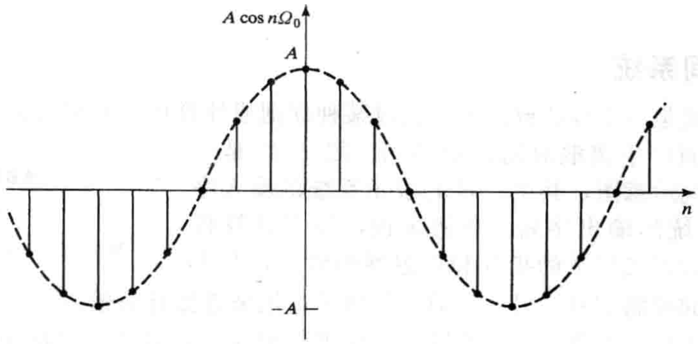


图 2.21　无阻尼离散时间正弦信号

$$C = Ae^{j\phi}$$
$$a = e^{\Sigma_0 + j\Omega_0} \tag{2.55}$$

因此 $x[n]$ 可表示为

$$x[n] = Ae^{j\phi}e^{\Sigma_0 n + jn\Omega_0} = Ae^{\Sigma_0 n}e^{j(n\Omega_0+\phi)} = Ae^{\Sigma_0 n}\cos(n\Omega_0+\phi) + jAe^{\Sigma_0 n}\sin(n\Omega_0+\phi) \tag{2.56}$$

式(2.56)中信号的实部如图 2.22 所示，其中图 2.22a 显示的是当 $\Sigma_0>0$ 的情况，图 2.22b 显示的是当 $\Sigma_0<0$ 的情况。正弦信号的轮廓由 $Ae^{\Sigma_0 n}$ 控制。

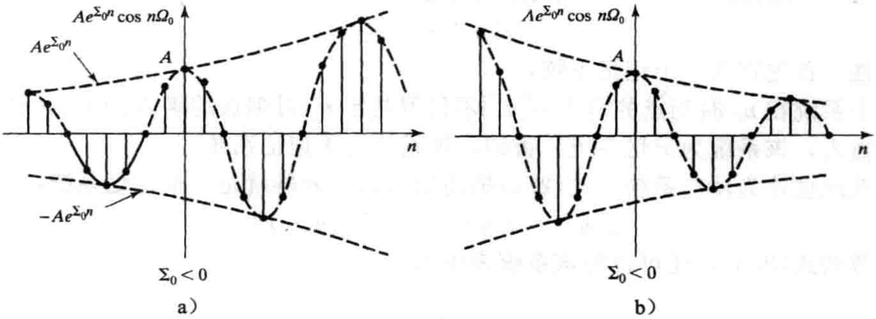


图 2.22　复指数信号的实部

再讨论式(2.52)中的指数信号，由 $\cos(n\pi)=(-1)^n$，可知

$$x[n] = Ca^n = C\alpha^n(-1)^n = C\alpha^n\cos(n\pi) \tag{2.57}$$

由此可以得出，当 a 为负实数时，信号 $x[n]=Ca^n$ 可以看作是对正弦信号在每个周期内抽样两个样点，其包络为指数函数。

在本节的最后，假设对信号 $x(t)=e^{\sigma_0 t}\cos(\omega_0 t)$ 进行抽样，可得

$$x[n] = e^{\sigma_0 t}\cos(\omega_0 t)\big|_{t=nT} = e^{\Sigma_0 n}\cos(n\Omega_0) \tag{2.58}$$

其与式(2.56)对比可以看出，连续时间信号的参数与离散时间信号的参数存在一定联系

$$\Sigma_0 = \sigma_0 T, \Omega_0 = \omega_0 T \tag{2.59}$$

因此可将式(2.56)中的信号看作是对一个连续时间复指数信号

$$x(t) = Ae^{(\sigma_0+j\omega_0)t}$$

进行抽样的结果，抽样周期为 T，其他参数如式(2.59)所示。

在本节中，多数情况下，离散时间信号可以看作由连续时间信号抽样得到。比如，数字滤波器的输入信号是通过对连续时间信号抽样得到。然而，很多情况下，离散时间信号也可以直接生成。比如，一个数字滤波器的输出信号是通过差分方程计算得来，并不是通

过抽样而得到。

2.5 离散时间系统

离散时间系统是一个算法或装置，通过某种规则或计算程序对输入的离散时间序列进行某种运算，从而产生离散时间输出序列。图 2.23 是一个简单的离散系统示意图，其中，$x[n]$表示系统的输入序列，$y[n]$表示系统的输出序列。举例来说，数字计算机(离散时间装置)收到飞行员的指令和传感器的输入，并生成控制飞机动作的控制信号。另一个简单的例子是用来近似计算输入信号积分的算法，输出为离散序列，比如 2.1 节中提到的欧拉方法进行积分。后面讨论的离散时间系统大多称为数字滤波器。

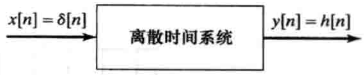


图 2.23 一个离散时间系统

离散时间系统通常用式(2.4)～式(2.7)中的差分方程描述。

离散时间系统也可以用其脉冲响应描述。一个系统的脉冲响应是指其输入信号为单位脉冲函数时系统的输出，如图 2.23 所示。

2.6 节将会详细讨论利用脉冲响应描述的离散时间系统，即系统的输入-输出关系完全由其脉冲响应得到。实际上，除了离散时间系统之外，所有系统都有各自的特点和性质，这些特性决定了它们的使用场合。下面介绍系统的某些性质。

2.5.1 离散时间系统的特性

定义 $x[n]$为系统的输入，$y[n]$为系统的输出，则输入-输出关系以符号简单表示为

$$x[n] \to y[n] \tag{2.60}$$

系统记忆性 首先定义一个记忆系统：

当一个系统在 n_0 时刻点的输出 $y[n_0]$不仅取决于 n_0 时刻点的输入 $x[n_0]$，还取决于其他时刻点输入，该系统为记忆系统。否则，该系统为无记忆系统。

记忆系统也称为动态系统。式(2.5)描述的欧拉积分器就是一个记忆系统：

$$y[n] = y[n-1] + Tx[n-1]$$

回忆 2.1 节的式(2.8)，还可以将该系统表示为

$$y[n] = T\sum_{k=-\infty}^{n-1} x[k] \tag{2.61}$$

可以看到系统当前时刻的输出由之前所有时刻的输入信号共同决定。

对于放大器

$$y[n] = Ax[n]$$

其输出信号仅由输入信号与常数相乘，因此该系统不是记忆系统，即当前输出仅取决于当前输入。

2.5.2 可逆性

下面定义可逆性。

可逆性 如果不同的输入信号得到不同的输出信号，则该系统是可逆的。

可逆系统的另一种定义为，其输入信号可以由其输出信号决定。比如，一个非记忆系统

$$y[n] = |x[n]|$$

是不可逆的。当输入为+2 和−2 时有同样的输出+2。

2.5.3 逆系统

逆系统与可逆性有关。

逆系统　一个系统 G 的逆系统 G_i，当其与原系统 G 级联时，整个系统为恒等系统。

恒等系统的定义为 $x[n]=y[n]$。

系统 G_i 是系统 G 的逆系统，当

$$G_i(G(x[n])) = x[n] \tag{2.62}$$

举例说明一个逆系统，比如，输入-输出关系如下的放大器

$$y[n] = G(x[n]) = Ax[n]$$

其逆系统为

$$G_i(y[n]) = A^{-1}y[n]$$

因此

$$G_i(G(x[n])) = A^{-1}(Ax[n]) = x[n]$$

2.5.4　因果性

无论是离散系统还是连续系统，所有物理上可实现的系统都是因果系统。

因果系统　因果系统是指系统输出只与系统当前输入以及过去的输入有关。

单位延时系统就是一个因果系统，如图 2.18a 所示。单位延时系统的输入为 $x[n]$，输出 $y[n]=x[n-1]$。平均滤波器也是一个因果系统，例如

$$y[n] = \frac{x[n]+x[n-1]+x[n-2]}{3}$$

该系统的输出取决于当前输入以及过去的输入。

非因果系统的实例为单位超前系统，即输入信号为 $x[n]$，输出信号为 $x[n+1]$。另一个非因果系统为式(2.63)表示的平均系统

$$y[n] = [x[n-1]+x[n]+x[n+1]]/3 \tag{2.63}$$

若想得到这个系统当前的输出 $y[n]$，需要知道系统未来的输入 $x[n+1]$。式(2.63)所示的滤波器只能应用于“离线”或者非实时情况，即通过将信号提前存入存储器以得到信号“未来”的输入值。

2.5.5　稳定性

离散时间系统的稳定性与连续时间系统稳定性的定义相同。

BIBO 稳定性　如果一个系统的输入有界，其输出也有界；则该系统是稳定系统。

下面给出有界输入/有界输出(BIBO)稳定性的具体表示形式，即存在正数 M 使得输入信号 $x[n]$ 满足

$$对所有\ n, |x[n]| \leqslant M \tag{2.64}$$

若存在一个数 R，使得系统输出满足

$$对所有\ n, |y[n]| \leqslant R \tag{2.65}$$

则该系统是 BIBO 稳定系统。R 通常是式(2.64)中 M 的函数。

注意，式(2.5)所示系统

$$y[n] = y[n-1] + Tx[n-1]$$

不是一个 BIBO 稳定系统，因为如果输入信号为常数 1，那么输出信号随着 n 的增大无限增大。

下面讨论系统

$$y[n] = n^2|x[n]|$$

其中，$x[n]=Mu[n]$，M 为有界常数。由于 $n\to\infty$时输入信号为常量 M，因而输入信号有界。但当 $n\to\infty$时 $y[n]\to\infty$，因此输出信号是无界的，该系统是不稳定系统。

下面讨论以差分方程描述的系统

$$y[n] - 0.9y[n-1] = 0.1x[n]$$

令 $y[0]=A$，$x[n]=Bu[n]$，A 和 B 均有界。则系统输出可以通过迭代得到

$$
\begin{aligned}
y[1] &= 0.1B+0.9A \\
y[2] &= 0.19B+0.81A \\
y[3] &= 0.271B+0.729A \\
y[4] &= 0.344B+0.656A \\
y[5] &= 0.410B+0.590A \\
&\vdots \\
y[9] &= 0.613B+0.387A \\
&\vdots
\end{aligned}
$$

可以看出随着 n 的增加，$y[n]\to B$，这说明该系统为 BIBO 稳定。第 3 章将介绍其他证明系统 BIBO 稳定的方法。

2.5.6　非时变性

离散时间系统的非时变性与连续时间系统的定义相同。

非时变系统　当一个系统的输入信号时移时，其输出信号也产生相同的时移，则该系统为非时变系统。

定义中离散时间增量 n 表示时间，对于一个非时变系统，其输入为 $x[n]$，输出为 $y[n]$，当输入为 $x[n-n_0]$时，系统输出为 $y[n-n_0]$。即

$$
\begin{aligned}
x[n] &\to y[n] \\
x[n-n_0] &\to y[n-n_0]
\end{aligned} \tag{2.66}
$$

系统的非时变性意味着系统本身并不随时间变化而改变。例如，如果银行在每月月初对已收存款支付固定的利息，也就是说，银行系统的响应不随时间变化而改变，则该系统是非时变系统。如果银行每月的利率都不同，那么本月银行系统的响应与下个月的就会不一样，系统就是时变系统。讨论以下由输入-输出关系描述的系统

$$y[n]=n^2x[n]$$

如果该系统的输入 $x[n]$延时 n_0，那么输出为 $n^2x[n-n_0]$，而输出信号延迟 n_0 时，$y[n-n_0]=(n-n_0)^2x[n-n_0]$。对于该系统，当输入信号延时为 $x[n-n_0]$时，输出信号并非为$y[n-n_0]$，因此，该系统是时变系统。

2.5.7　线性系统

系统的线性特性是最重要的特性之一，同样先定义系统的输入信号为 $x[n]$，输出信号为 $y[n]$。

线性系统　线性系统满足如下两条准则：

1. 叠加性。如果 $x_1[n]\to y_1[n]$，$x_2[n]\to y_2[n]$，那么

$$x_1[n]+x_2[n]\to y_1[n]+y_2[n] \tag{2.67}$$

2. 均匀性。如果 $x[n]\to y[n]$，则对于常数 a 有

$$ax[n]=ay[n] \tag{2.68}$$

这条准则对所有的 $x[n]$和 a 都适用。

以上两条准则组成了叠加原理，表示为

$$a_1x_1[n]+a_2x_2[n]\to a_1y_1[n]+a_2y_2[n] \tag{2.69}$$

其中，a_1 和 a_2 为任意常数。满足叠加原理的系统就是线性系统。

下面讨论式(2.8)中的欧拉积分器。

$$y[n]=T\sum_{k=-\infty}^{n-1}x[k]$$

输入信号为 $x_1[n]$时，$y_1[n]=T\sum_{k=-\infty}^{n-1}x_1[k]$；输入信号为 $x_2[n]$时，$y_2[n]=T\sum_{k=-\infty}^{n-1}x_2[k]$；输入为 $a_1x_1[n]+a_2x_2[n]$时，$y[n]=a_1T\sum_{k=-\infty}^{n-1}x_1[k]+a_2T\sum_{k=-\infty}^{n-1}x_2[k]=a_1y_1[n]+a_2y_2[n]$。由于该系统满足叠加原理，因而是线性系统。

举一个非线性系统的实例。系统 $y[n]=x[n]^2$，有 $y_1[n]=x_1[n]^2$，$y_2[n]=x_2[n]^2$，对于输入 $a_1x_1[n]+a_2x_2[n]$，输出

$$y[n]=(a_1x_1[n]+a_2x_2[n])^2=a_1^2x_1[n]^2+2a_1a_2x_1[n]x_2[n]+a_2^2x_2[n]^2$$

并不等于 $a_1y_1[n]+a_2y_2[n]$，因此该系统不满足叠加定理，不是线性系统。

LTI 系统 线性非时变系统(LTI 系统)是线性系统，同时也是非时变系统。离散 LTI 系统可以通过常系数线性差分方程来表示。式(2.6)和式(2.7)给出了 n 阶常系数线性差分方程的一般形式，即

$$\begin{aligned}y[n]=&a_1y[n-1]+a_2y[n-2]+\cdots+a_Ny[n-N]+b_0x[n]+b_1x[n-1]\\&+b_2x[n-2]+\cdots+b_Mx[n-M]\end{aligned}$$

该方程称为 N 阶差分方程，另一种表达方式是将 n 替换成 $n+N$。

2.6 离散时间系统的卷积

本节只对 LTI 系统进行讨论，也就是线性非时变离散时间系统。着重讨论该系统的原因为：

1. 很多物理系统都可以准确表示为 LTI 系统，比如，许多数字滤波器都可设计为线性非时变系统。

2. 无论连续时间还是离散时间系统，LTI 系统方程可以通过数学方法准确求解，而目前还没有能够求解非 LTI 系统的通用方法。

3. 对 LTI 系统设计和分析的研究已经成熟，在 LTI 数字滤波器设计方面尤其如此。

2.6.1 离散时间信号的脉冲序列表示

本节将通过脉冲函数来表示一个信号 $x[n]$，这种表示形式在研究离散时间 LTI 系统的特性时十分有用。

首先回忆一下离散时间脉冲函数(也称单位样点函数)的定义

$$\delta[n-n_0]=\begin{cases}1, & n=n_0\\0, & n\neq n_0\end{cases}\tag{2.70}$$

即当自变量为 0 时，其函数值为 1；否则，其函数值恒为 0。由此可得

$$x[n]\delta[n-n_0]=x[n_0]\delta[n-n_0]$$

对于图 2.24a 中的信号，为简化运算只赋予该信号三个非零值。利用式(2.70)可得

$$x_{-1}[n]=x[n]\delta[n+1]=x[-1]\delta[n+1]=\begin{cases}x[-1], & n=-1\\0, & n\neq-1\end{cases}$$

同理

$$x_0[n]=x[n]\delta[n]=x[0]\delta[n]=\begin{cases}x[0], & n=0\\0, & n\neq 0\end{cases}$$

$$x_1[n]=x[n]\delta[n-1]=x[1]\delta[n-1]=\begin{cases}x[1], & n=1\\0, & n\neq 1\end{cases}$$

图 2.24 给出了这三个信号。其中 $x[n]$的 3 个非零值都对应地存在于这 3 个信号中，因此可将 $x[n]$表示为

$$x[n]=x_{-1}[n]+x_0[n]+x_1[n]=x[-1]\delta[n+1]+x[0]\delta[n]+x[1]\delta[n-1]$$

$$= \sum_{k=-1}^{1} x[k]\delta[n-k] \tag{2.71}$$

将结果进行推广，可得

$$x[k]\delta[n-k] = \begin{cases} x[k], & n = k \\ 0, & n \neq k \end{cases} \tag{2.72}$$

由此可以得到信号 $x[n]$

$$x[n] = \sum_{k=-\infty}^{\infty} x[k]\delta[n-k] \tag{2.73}$$

这种信号表达方式将会在下一节中起到重要作用。

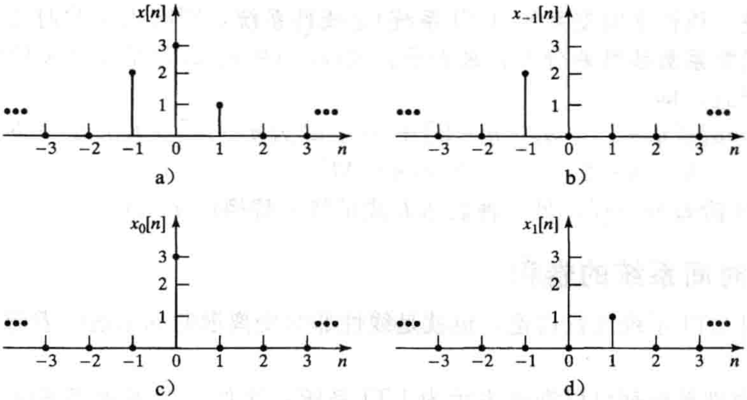


图 2.24　一个信号以离散时间脉冲函数表示

函数 $\delta[n]$称作单位样点函数或者单位脉冲函数。我们使用单位脉冲函数以体现其连续与离散之间的对称性(离散时间脉冲函数和系统，连续时间脉冲函数和系统)。

2.6.2　卷积

接下来讨论离散 LTI 系统输入与输出之间的关系式。假设对于图 2.25 中的系统，输入为单位脉冲函数 $\delta[n]$，该函数仅在 $n=0$ 时取值为 1，其他时刻值为 0。

对于单位脉冲输入，图 2.25 给出了 LTI 系统的响应 $h[n]$，即

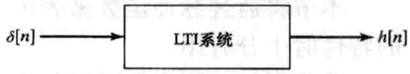


图 2.25　LTI 系统的脉冲响应

$$\delta[n] \rightarrow h[n] \tag{2.74}$$

另外由于该系统是非时变系统，

$$\delta[n-k] \rightarrow h[n-k] \tag{2.75}$$

其中，$h[\cdot]$表示系统的单位脉冲响应。又因为该系统为线性系统，所以可在式(2.75)两边同时乘以 $x[k]$，得到

$$x[k]\delta[n-k] \rightarrow x[k]h[n-k] \tag{2.76}$$

前面已经介绍过，输入信号 $x[n]$可以表示为脉冲函数之和的形式：

$$x[n] = \sum_{k=-\infty}^{\infty} x[k]\delta[n-k]$$

根据式(2.76)，当输入信号是脉冲函数的累加和时，其对应的输出信号也是脉冲函数的累加，根据式(2.69)的线性特性，输入-输出关系满足

$$x[n] = \sum_{k=-\infty}^{\infty} x[k]\delta[n-k] \rightarrow y[n] = \sum_{k=-\infty}^{\infty} x[k]h[n-k] \tag{2.77}$$

该结果称为卷积和，是离散时间 LTI 系统的基本结论。用星号来表示这种求和

$$y[n]=\sum_{k=-\infty}^{\infty}x[k]h[n-k]=x[n]*h[n] \tag{2.78}$$

为了说明卷积和的性质，计算 $n=0$ 时系统的输出：

$$y[0]=\cdots+x[-2]h[2]+x[-1]h[1]+x[0]h[0]+x[1]h[-1]+x[2]h[-2]+\cdots \tag{2.79}$$

注意，在计算卷积和的每个数值时，随着输入信号 $x[\cdot]$参与计算的项逐渐增加，$h[\cdot]$参与计算的项逐渐减小。$y[n]$的一般形式为

$$\begin{aligned}y[n]=&\cdots+x[-2]h[n+2]+x[-1]h[n+1]+x[0]h[n]\\&+x[1]h[n-1]+\cdots\end{aligned} \tag{2.80}$$

可以看到式(2.80)中 $x[\cdot]$和 $h[\cdot]$是对称出现的，因此卷积和可以有两种表示方式：

$$y[n]=\sum_{k=-\infty}^{\infty}x[k]h[n-k]=x[n]*h[n]=\sum_{k=-\infty}^{\infty}h[k]x[n-k]=h[n]*x[n] \tag{2.81}$$

第二种表示方式可由第一种表示方式替换变量得到。

下面介绍卷积和的两个性质。对于式(2.80)中的 $y[n]$，其累加项中的每项$x[k_1]h[k_2]$都存在$(k_1+k_2)=n$。比如，计算 $y[5]$时，$x[10]h[-5]$是累加项，而 $x[10]h[-6]$则不是累加项。

卷积和的第二个特性可以通过令 $x[n]=\delta[n]$导出。有

$$y[n]=\delta[n]*h[n]=h[n] \tag{2.82}$$

输出是脉冲响应。将 $h[n]$中的 n 替换成$(n-n_0)$可得

$$\delta[n]*h[n-n_0]=h[n-n_0]$$

又由于系统的非时变性，可以得到式(2.82)的一般表示形式

$$\delta[n-n_0]*h[n]=h[n-n_0]$$

由上面两个等式可以得到

$$\delta[n]*h[n-n_0]=h[n-n_0]=\delta[n-n_0]*h[n]$$

这里注意，等式对 $h[n]$没有任何限制，因此可将 $h[n]$替换成一般函数 $g[n]$，进而得到了卷积和的第二个特性：

$$\delta[n]*g[n-n_0]=\delta[n-n_0]*g[n]=g[n-n_0] \tag{2.83}$$

特别注意不要将卷积和与乘积运算的符号混淆。脉冲函数的乘法特性表现为

$$\delta[n]g[n-n_0]=g[-n_0]\delta[n]$$

以及

$$\delta[n-n_0]g[n]=g[n_0]\delta[n-n_0]$$

因为当$-\infty<n<\infty$时，仅有 $\delta[0]$为非零值。

根据卷积和式(2.81)可以看出，对于已知的 $h[n]$，可计算任意输入信号经过 LTI 离散系统的输出响应。因此，一个离散 LTI 系统的脉冲响应 $h[n]$能够完全描述该系统的输入-输出关系。下面给出两个应用卷积和的具体实例。

例 2-9 有限脉冲响应系统

讨论图 2.26 所示系统，其中标有“T”的模块为单位延迟，根据此图可以直接得到描述该系统的差分方程：

$$y[n]=(x[n]+x[n-1]+x[n-2])/3 \tag{2.84}$$

该系统的输出是过去 3 个输入的平均数，是一个滑动平均滤波器，有广泛的应用。令系统输入 $x[n]=\delta[n]$，进而得到系统的脉冲响应 $h[n]$：

$$y[n]=h[n]=(\delta[n]+\delta[n-1]+\delta[n-2])/3 \tag{2.85}$$

因此有

$$h[0] = (\delta[n] + \delta[n-1] + \delta[n-2])/3\,|_{n=0} = (1+0+0)/3 = 1/3$$
$$h[1] = (\delta[n] + \delta[n-1] + \delta[n-2])/3\,|_{n=1} = (0+1+0)/3 = 1/3$$
$$h[2] = (\delta[n] + \delta[n-1] + \delta[n-2])/3\,|_{n=2} = (0+0+1)/3 = 1/3$$
$$h[n] = 0, n\text{ 为其他值}$$

这是一个有限脉冲响应(FIR)系统，即系统的脉冲响应包含有限个非零值。作为一个练习，读者可以对图2.26中的系统进行输入信号$x[n]=\delta[n]$的跟踪，来验证得到的响应$h[n]$。(存储在两个延迟器中的初始数据应当是0，否则在脉冲响应之外将叠加有零输入响应。) ◀

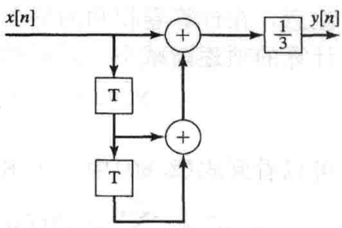


图2.26　例2-9的系统

例2-10　利用卷积求解LTI系统的响应

继续利用例2-9中的系统来阐述卷积和的应用，令系统输入为：

$$x[1] = 3; x[2] = 4.5$$
$$x[3] = 6; x[n] = 0, n\text{ 为其他值时}$$

图2.27a给出了$x[n]$的图形。由式(2.81)可得响应$y[n]$为

$$y[n] = \sum_{k=-\infty}^{\infty} x[n-k]h[k]$$

下面绘制$x[n-k]$的图形。首先将时间变量n替换成m，如图2.27a所示，利用$x[m]$使得接下来的绘制更加方便。介绍卷积和时，将n看作常量，因此根据2.2节的内容可得

$$x_t[k] = x[m]\,|_{m=n-k} = x[n-k]$$

因此　　　$$m = n-k \Rightarrow k = n-m$$

接着，在图2.27a中的m轴下方绘制k轴，得到图2.27b中$x[n-k]$的图形。卷积和中的第二项$h[k]$可以由例2-9的结果得到，如图2.27a所示。

首先，在图2.27c中给出了当$n=0$时$x[n-k]$和$h[k]$的图形。可以看到两个信号的乘积$x[n-k]h[k]$为0，即当$n=0$，$y[n]=0$。另外，因为信号$x[n-k]$随着$n<0$向左平移，可以看到当$n<0$时$y[n]=0$。

图2.27d给出了当$n=1$时$x[n-k]$和$h[k]$的图形。

$$x[n-k]h[k]\,|_{n=1,k=0} = x[1]h[0] = \frac{1}{3}\times 3 = 1$$

而且当k为其他值时，乘积项为0，因此$y[1]=1$。

图2.27e给出了当$n=2$时$x[n-k]$和$h[k]$的状态，卷积中只有$k=0$和$k=1$的乘积项不为0，所有其他乘积项都等于零，因此有

$$y[2] = x[2]h[0] + x[1]h[1] = 1.5 + 1 = 2.5$$

同理

$$y[3] = \frac{1}{3}\times 6 + \frac{1}{3}\times 4.5 + \frac{1}{3}\times 3 = 4.5$$

$$y[4] = \frac{1}{3}\times 6 + \frac{1}{3}\times 4.5 + 0\times 3 = 3.5$$

$$y[5] = \frac{1}{3}\times 6 + 0\times 4.5 + 0\times 3 = 2$$

当$n>5$时，$y[n]=0$。因此输出为

$$y[n] = \delta[n-1] + 2.5\delta[n-2] + 4.5\delta[n-3] + 3.5\delta[n-4] + 2\delta[n-5]$$

输出信号$y[n]$如图2.27f所示。

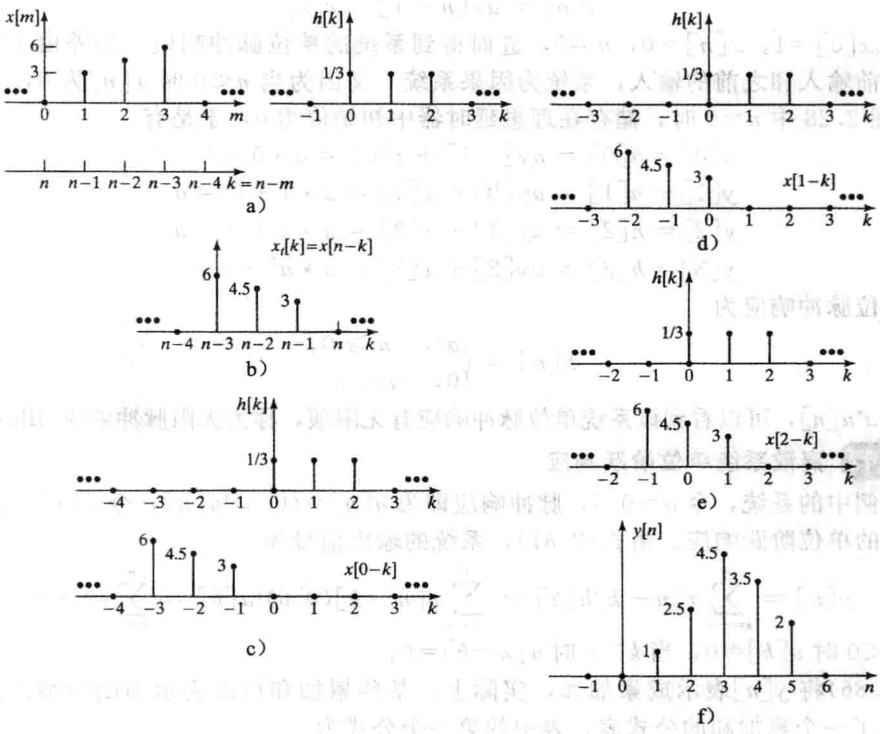


图 2.27　例 2-10 中的信号

另外，输出信号可以通过卷积和展开来计算得到：

$$
\begin{aligned}
y[n] &= \cdots + x[n-3]h[3] + x[n-2]h[2] \\
&\quad + x[n-1]h[1] + x[n]h[0] + x[n+1]h[-1] + \cdots \\
&= x[n-2]h[2] + x[n-1]h[1] + x[0]h[0]
\end{aligned}
$$

由于 $h[n]$ 只在 $n=0$，1，2 时是非零的，因此展开式中仅有 3 个非零项。如例 2-9 所示，输入信号也可以通过系统进行跟踪作为第三种检验方法。最后可以通过如下 MATLAB 程序进行验证：

```
n=1:5;
x=[3 4.5 6];
h=[1/3 1/3 1/3];
y=conv(x,h)
stem(n,y,'fill'), grid
```

注意，在例 2-9 和例 2-10 中，我们给出了三种系统的描述方法：

1. 式(2.85)中的脉冲响应；
2. 式(2.84)中的差分方程；
3. 图(2.26)中的系统框图。

实际应用中，描述一个离散时间系统的框图和差分方程都应该在技术文档中给出，但系统的脉冲响应很少直接给出，一般给出系统脉冲响应的 z 变换，第 3 章会具体介绍其内容。

例 2-10 介绍的系统是有限脉冲响应，而下面例子中的系统具有无限脉冲响应。

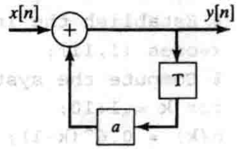


图 2.28　例 2-11 的系统

例 2-11　计算一个离散系统的脉冲响应

考虑图 2.28 中的系统，首先根据图 2.28 直接写出系统的差分方程：

$$y[n]=ay[n-1]+x[n]$$

令输入为 $x[0]=1$，$x[n]=0$，$n\neq 0$，进而得到系统的单位脉冲响应。另外由于系统输出取决于当前输入和之前的输入，系统为因果系统。又因为当 $n<0$ 时 $x[n]$ 为 0，$y[n]$ 也为 0。所以图 2.28 中 $n=0$ 时，储存在理想延时器中初始值为 0，于是有

$$y[0]=h[0]=ay[-1]+x[0]=a\cdot 0+1=1$$
$$y[1]=h[1]=ay[0]+x[1]=a\cdot 1+0=a$$
$$y[2]=h[2]=ay[1]+x[2]=a\cdot a+0=a^2$$
$$y[3]=h[3]=ay[2]+x[3]=a\cdot a^2+0=a^3$$

系统的单位脉冲响应为

$$h[n]=\begin{cases}a^n, & n\geqslant 0,\\ 0, & n<0\end{cases}$$

或 $h[n]=a^nu[n]$，可以看到该系统单位脉冲响应有无限项，称为无限脉冲响应(IIR)系统。◀

例 2-12 离散系统单位阶跃响应

对上例中的系统，令 $a=0.6$，脉冲响应即为 $h[n]=(0.6)^nu[n]$，令 $x[n]=u[n]$即可得到系统的单位阶跃响应。由式(2.81)，系统的输出信号为

$$y[n]=\sum_{k=-\infty}^{\infty}x[n-k]h[k]=\sum_{k=-\infty}^{\infty}u[n-k](0.6)^ku[k]=\sum_{k=0}^{n}(0.6)^k \tag{2.86}$$

因为当 $k<0$ 时 $u[k]=0$，当 $k>n$ 时 $u[n-k]=0$。

式(2.86)将 $y[n]$表示成累加和，实际上，某些累加和可以表示为闭合解的形式。附录 C 给出了一个累加和的公式表，表中的第一个公式为

$$\sum_{k=0}^{n}a^k=\frac{1-a^{n+1}}{1-a} \tag{2.87}$$

注意，当 $a=1$ 时等式右侧无定义，但根据式(2.87)左侧项，累加和中 $a=1$ 时结果为$(n+1)$。最终根据式(2.86)和式(2.87)，可得

$$y[n]=\sum_{k=0}^{n}(0.6)^k=\frac{1-(0.6)^{n+1}}{1-0.6}=2.5[1-(0.6)^{n+1}],n\geqslant 0$$

将 $y[n]$的值展开

$y[0]=1$， $y[5]=2.383$

$y[1]=1.6$， ⋮

$y[2]=1.96$， $y[10]=2.485$

$y[3]=2.176$， ⋮

$y[4]=2.306$， $y[\infty]=2.5$

可以看到 $y[n]$达到稳定状态时的值为 2.5。图 2.29给出了 $y[n]$的图形，可以看到该响应是按指数规律变化的。(2.4 节介绍了信号 a^n 为指数信号。) ◀

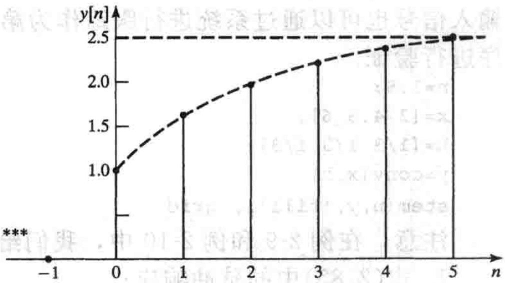


图 2.29 例 2-12 的系统响应

最后可通过如下 MATLAB 程序进行验证：

```
% This MATLAB program performs the convolution operation of
% Example 2.12.
% Establish the input unit-step function.
x=ones (1,11);
% Compute the system impulse response.
for k = 1:10;
h(k) = 0.6^(k-1);
end
```

```
% Convolve the input with the impulse response and plot.
c=conv (x,h);
for k=1:11;
n(k)=k-1;
y(k)=c(k);
end
[n' y']
stem(n, y, 'fill')
```

2.6.3 卷积的性质

下面讨论三条与系统相关的卷积和性质。

1. **交换律**。卷积和对于 $x[n]$和 $h[n]$是对称的：

$$x[n]*h[n]=h[n]*x[n] \tag{2.88}$$

式(2.81)给出了证明，图 2.30 给出了图形解释，其中一个 LTI 系统由标有脉冲响应的模块表示，由式(2.88)，两个系统的输出是完全相同的。



图 2.30　交换性

2. **结合律**。三个信号进行卷积时，最后结果与卷积的顺序无关。比如

$$(f[n]*g[n])*h[n]=f[n]*(g[n]*h[n])=(h[n]*f[n])*g[n] \tag{2.89}$$

该性质的证明并没有给出。

举例说明这条性质，对于图 2.31a 中所示系统，其输出为：

$$y[n]=y_1[n]*h_2[n]=(x[n]*h_1[n])*h_2[n]$$

根据结合律，

$$(x[n]*h_1[n])*h_2[n]=x[n]*(h_1[n]*h_2[n])=(x[n]*h_2[n])*h_1[n] \tag{2.90}$$

因此，图 2.31a 中的两个系统顺序变换后并不改变输入-输出关系。

另外根据式(2.90)，图 2.31a 中的两个级联系统可以替换成一个系统，该系统的脉冲响应为

$$h[n]=h_1[n]*h_2[n] \tag{2.91}$$

这样可以更清晰地体现系统的输入-输出关系。图 2.31b 给出了这条性质的图示。同时结合律适用于 m 个级联的 LTI 系统，其脉冲响应为

$$h[n]=h_1[n]*h_2[n]*\cdots*h_m[n]$$

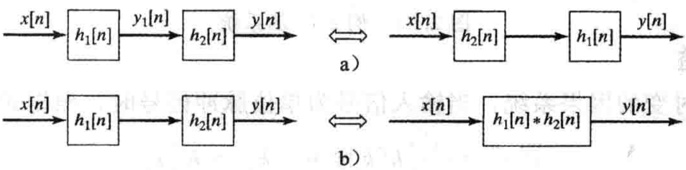


图 2.31　结合性

3. **分配律**。卷积和满足如下关系：

$$x[n]*h_1[n]+x[n]*h_2[n]=x[n]*(h_1[n]+h_2[n]) \tag{2.92}$$

利用式(2.78)证明分配律：

$$\begin{aligned}x[n]*h_1[n]+x[n]*h_2[n]&=\sum_{k=-\infty}^{\infty}x[k]h_1[n-k]+\sum_{k=-\infty}^{\infty}x[k]h_2[n-k]\\&=\sum_{k=-\infty}^{\infty}x[k](h_1[n-k]+h_2[n-k])\\&=x[n]*(h_1[n]+h_2[n])\end{aligned}$$

图 2.32 中两个并联的系统给出了分配律的图示，系统输出为：

$$y[n]=x[n]*h_1[n]+x[n]*h_2[n]=x[n]*(h_1[n]+h_2[n]) \tag{2.93}$$

因此整个系统的脉冲响应为子系统脉冲响应的和：

$$h[n]=h_1[n]+h_2[n] \tag{2.94}$$

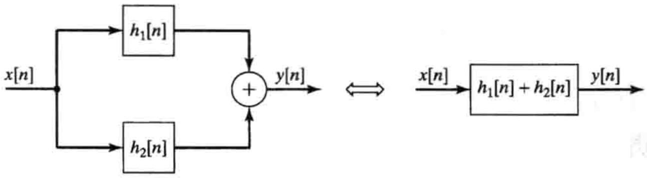


图 2.32 分配律

下面给出应用上述性质的实例。

例 2-13 互连系统的脉冲响应

若要得到图 2.33a 中所示系统的脉冲响应，就需先求出其子系统的脉冲响应。首先，级联系统 1 和系统 2 的脉冲响应为

$$h_a[n]=h_1[n]*h_2[n]$$

如图 2.33b 所示，并联的系统 a 和系统 3 的脉冲响应为

$$h_b[n]=h_a[n]+h_3[n]=h_1[n]*h_2[n]+h_3[n]$$

如图 2.33c 所示，将得到的系统 b 与系统 4 进行级联，进而得到图 2.33d 所示系统的脉冲响应：

$$h[n]=h_b[n]*h_4[n]=(h_1[n]*h_2[n]+h_3[n])*h_4[n]$$ ◀

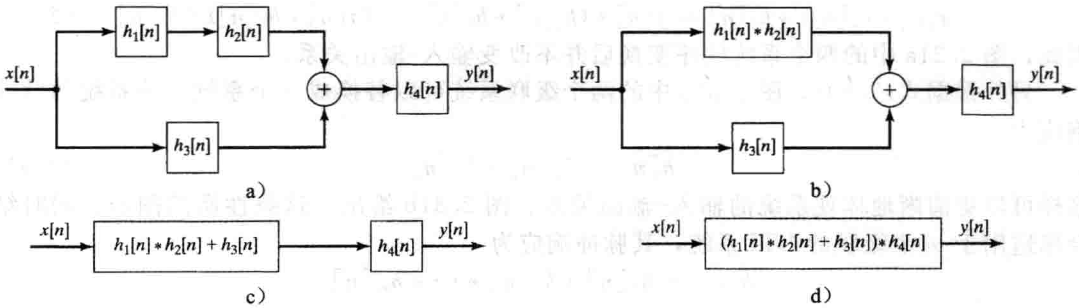


图 2.33 例 2-13 的系统

2.6.4 功率增益

对于线性非时变的因果系统，当输入信号为单位脉冲信号时，根据式(2.81)有：

$$y[n]=\sum_{k=0}^{\infty}h[k]\delta[n-k]=h[n] \tag{2.95}$$

输入信号连续 N 个值的归一化平均功率为

$$P_x=\frac{1}{N}\sum_{n=0}^{N-1}|x[n]|^2=\frac{1}{N}\sum_{n=0}^{N}|\delta[n]|^2=\frac{1}{N}$$

输出信号连续 N 个值的归一化平均功率为

$$P_y=\frac{1}{N}\sum_{n=0}^{N-1}|y[n]|^2=\frac{1}{N}\sum_{n=0}^{N}|h[n]|^2 \tag{2.96}$$

由此可得系统的功率增益为

$$\frac{\text{输出信号功率}}{\text{输入信号功率}}=\frac{P_y}{P_x}=\sum_{n=0}^{N}|h[n]|^2 \tag{2.97}$$

因此，如果一个离散时间 LTI 系统的脉冲响应是已知的，那么该系统的功率增益就可以根据式(2.97)计算得到。

根据本节内容可知，当一个离散 LTI 系统的脉冲响应已知时，该系统的响应就可由卷积和式(2.81)计算。因此，系统的输入-输出关系完全可以用该系统对脉冲函数的响应来描述。此外，本节还介绍了一种计算由子系统组成的 LTI 系统脉冲响应的方法，前提是已知这些子系统各自的脉冲响应。当然，还有一种等效但更加简便的计算方法就是利用子系统输入-输出特性，在变换域进行计算，将在第 3 章学习 z 变换时具体介绍。

小结

本章介绍了离散时间信号和系统，其是后续学习数字滤波器设计的理论基础。对于一个离散时间信号 $x[n]$来说，离散时间增量 n 表示当前的样点时刻。对于周期抽样后的信号，用表达式 $x[n]=x(nT)$表示，其中 T 是抽样周期。

本章介绍了离散时间信号的时间和幅度变换，这些变换包括时间和幅度的翻转、展缩和平移。

另外，本章也讨论了信号的奇偶性以及周期性等内容，可以看出离散时间周期信号比连续时间周期信号更加复杂。

本章还定义和分析了实际的物理系统中经常遇到的几种离散时间信号的数学模型，其中离散时间脉冲函数被定义为一个普通函数，这一点与连续时间冲激函数有所不同。

本章定义了离散时间系统的一些特性，如线性、非时变性、因果性和稳定性。

最后介绍了离散时间系统的脉冲响应，以及利用输入信号与系统脉冲响应的卷积和计算输出信号的方法。其中还给出了卷积的一些性质，如交换律、结合律和分配律。最后一节介绍了利用线性非时变系统的脉冲响应来计算该系统的功率增益的方法。

若想对本章内容进行更加深入学习，建议阅读附录 B 中参考文献 1 和 2。

习题

概念题

1. 什么情况下 $x[n]=x(nT)$？什么情况下不等于？
2. 如果对一个连续时间信号进行抽样，得到的抽样值输入数字计算机中，为什么通常计算机中显示的抽样值与原始信号抽样点处的值不同？
3. 讨论利用连续时间信号的抽样值计算积分时，抽样周期与计算精度之间的关系。
4. 讨论连续时间函数 $x(t)=u(t)-u(t-t_0)$与离散时间函数 $x[n]=u[n]-u[n-n_0]$的区别。
5. 讨论连续时间冲激函数 $\delta(t-t_0)$与离散时间脉冲函数 $\delta[n-n_0]$的区别。
6. 解释为什么对一个周期连续时间信号抽样不一定能够得到一个周期离散时间信号。该离散时间信号是周期信号的条件是什么？
7. 为什么一个信号的时间平均值始终取决于信号的偶分量？
8. 给出一个实例，连续时间周期信号经抽样后，得到的是离散时间非周期信号，抽样周期 $T=1\text{ms}$。
9. 用语言描述当一个信号经过系统时，信号卷积的过程。
10. 给出一个与例 2-9 中系统不同的有限脉冲响应系统。

分析设计题

2.1 节

2.1　下面哪个离散时间函数与其余的不同。

(a) $x_a[n]=u[n+1]-u[n-2]$

(b) $x_b[n]=\sum_{k=-1}^{1}\delta[n-k]$

(c) $x_c[n]=\begin{cases}1, & n\in[-1,1]\\ 0, & 其他\end{cases}$

(d) $x_d[n]=\delta[n+1]+\delta[n]+\delta[n-1]+\delta[n-2]$

2.2　写出 MATLAB 程序，生成 $u[n-4]-u[n-10]$在 $0\leqslant n\leqslant 15$ 时的图形。

2.3　图 P2.3 给出了数值积分计算中的梯形法则。nT 点

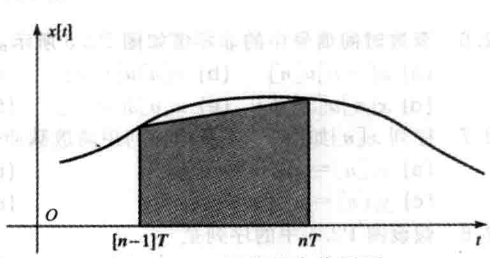


图 P2.3　梯形积分波形图

的积分值等于$(n-1)T$点积分值加上图 P2.3 中所示梯形的面积。

(a) 写出计算积分时表示$y[n]$与$x[n]$之间关系的差分方程。

(b) 写出 MATLAB 程序，计算当抽样周期 $T=0.1$s 时，信号 e^{-t}在 $0\leqslant t\leqslant 5$ 上的积分。

2.4　图 P2.4 所示序列由连续时间信号 $x(t)=e^{6.9315t}[u(t)-u(t-0.55)]$以抽样周期为 0.1 秒抽样得到。

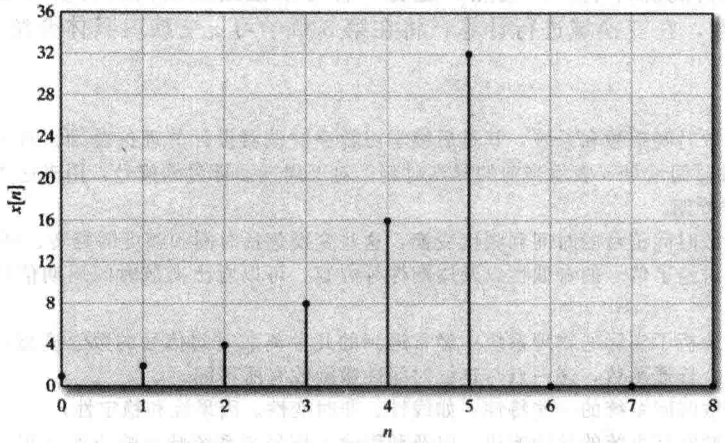


图 P2.4　离散时间抽样数据序列

(a) 利用矩形(欧拉)方法近似计算信号的积分。

(b) 利用梯形方法近似计算信号的积分。

(c) 计算连续时间信号 $x(t)$的积分。

(d) 比较(a)、(b)和(c)的结果。

2.2 节

2.5　离散时间信号中的非零值如图 P2.5 所示。请绘制如下信号：

(a) $x[2n]$　(b) $x[n/2]$　(c) $x[-n]$　(d) $x[-2n]$

(e) $x[n-2]$　(f) $x[2-n]$　(g) $2-3x[n]$　(h) $1+2x[n-2]$

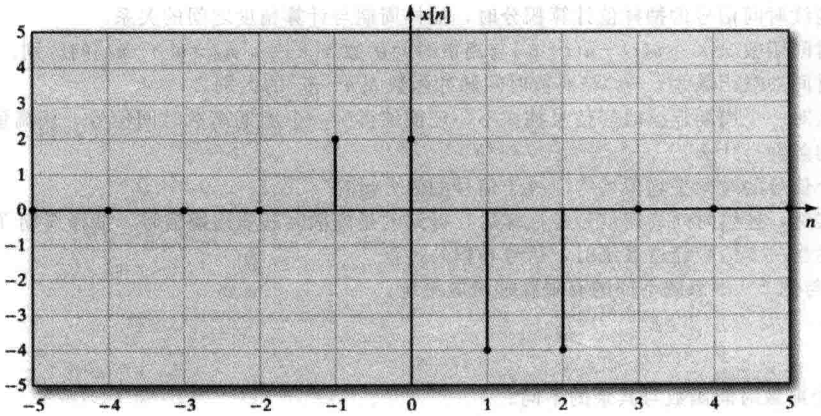


图 P2.5　离散时间序列

2.6　离散时间信号中的非零值如图 P2.5 所示。请绘制如下信号：

(a) $x[-n]u[n]$　(b) $x[n]u[-n]$　(c) $x[n]u[n-2]$

(d) $x[n]u[2-n]$　(e) $x[n]\delta[n-1]$　(f) $x[n](\delta[n]+\delta[n-2])$

2.7　序列 $x[n]$如图 P2.4 所示。利用离散脉冲信号描述下列序列，并绘制其草图。

(a) $y_a[n]=x[n-2]$ vs. n.　(b) $y_b[n]=x[n+3]$ vs. n.

(c) $y_c[n]=x[-n]$ vs. n.　(d) $y_d[n]=x[-n+3]$ vs. n

2.8　假设图 P2.5 中的序列是

$$x[n]=3v[0.5n-2]+2$$

求 $v[n]$并画出草图。

2.9　描述信号的时间和幅度变换的一般形式表示为

$$y[n]=Ax[an+b]+B$$

其中，a 是有理数，b 是整数。将该方程改写为以 $y[n]$表示 $x[n]$的形式。

2.3 节

2.10　证明下列结论：

(a) 两个偶函数的和是偶函数。

(b) 两个奇函数的和是奇函数。

(c) 一个奇函数和一个偶函数的和，既不是奇函数也不是偶函数。

(d) 两个奇函数的乘积是偶函数。

(e) 两个偶函数的乘积是偶函数。

(f) 一个奇函数和一个偶函数的乘积是奇函数。

2.11　(a) 判断下信号的奇偶性。

(i) $x_1[n]=3u[n-2]$　(ii) $x_2[n]=n$

(iii) $x_3[n]=(0.7)^{|n|}$　(iv) $x_4[n]=3+(0.7)^n+(0.7)^{-n}$

(v) $x_5[n]=\cos(n)$　(vi) $x_6[n]=\cos\left(n-\frac{\pi}{6}\right)$

(b) 画出 a 中信号的草图并验证结果。

(c) 计算每个信号的奇分量和偶分量。

2.12　(a) 分别计算并画出图 P2.12 中离散时间信号 $x[n]$的奇分量和偶分量。

(b) 写一段 MATLAB 程序，画出图 P2.12 中离散时间信号 $x[n]$的奇分量和偶分量。

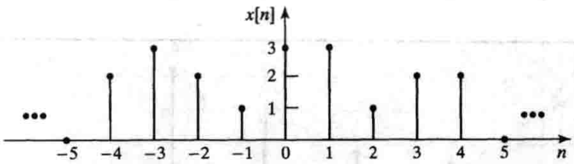


图 P2.12　离散时间信号

2.13　信号和的表达式为：$x_1=\cos(0.2\pi n)$，$x_2=\cos(0.125\pi n)$

(a) 判断两个信号是否为周期信号。如果是，则给出一个周期内抽样点的个数。

(b) 判断两个信号的和是否为周期信号。如果是，则给出一个周期内抽样点的个数。

2.14　(a)对连续时间信号 $x(t)=\cos(2\pi B)$每隔 T 秒进行抽样。判断当 T 为下列值时，所得离散时间信号 $x[n]=x(nT)$是否为周期信号：

(i) $T=5\text{s}$　(ii) $T=0.125\text{s}$　(iii) 0.1s

(iv) $T=0.130\text{s}$　(v) $T=\frac{4}{3}\text{s}$

(b) 对于(a)中的 5 种情况，当抽样后的信号为周期信号时，找出 $x[n]$一个周期内信号 $x(t)$的周期个数。

(c) 对于(a)中的 5 种情况，当抽样后的信号为周期信号时，找出 $x[n]$一个周期内抽样点的个数。

2.15　(a) 判断下列哪些离散时间信号是周期的。

(i) $\cos(\pi n)$　(ii) $\cos(3\pi n/2+\pi/4)$　(iii) $\sin(0.01\pi n)$

(iv) $\sin(3.15n)$　(v) $\sin(3.15\pi n)$　(vi) $\cos(\pi n/2)+1$

(b) 对于(a)中的周期信号，确定其每个周期的抽样个数。

2.4 节

2.16　某离散时间信号的表达式为 $x[n]=\begin{cases}0.5^n, & -6<n<6\\ 0, & \text{其他}\end{cases}$

(a) 以 n 为变量绘制序列的图形。

(b) 绘制 $x[n]u[n]$的图形。

(c) 绘制 $x[n]u[-n]$的图形。

(d) 绘制 $x[-n]u[-n]$的图形。

(e) 绘制 $x[2-n]u[2-n]$的图形。

2.17 下列离散时间信号都是以 $T=0.1\text{s}$ 为抽样周期对连续时间信号进行抽样得到。求解每个信号的时间常数 τ 以及每个正弦信号的频率 ω。

(a) $(0.3)^n$　　(b) $(0.3)^n\cos(n)$

(c) $(-0.3)^n$　　(d) $(0.3)^n\sin(n+1)$

(e) $(0.5)^n\cos(3n)$　　(f) $(0.5)^n\sin\left(120\pi n+\frac{\pi}{4}\right)$

2.18 利用 MATLAB 生成并绘制习题 2.17(a)～(f)的信号图形。

2.19 离散时间信号 $x[n]=4(0.6)^n\cos(4n)$由连续信号 $x(t)$以每秒 10 点抽样得到。

(a) 信号 $x(t)$的时间常数是多少?

(b) 信号 $x(t)$的振荡频率是多少?

(c) 使用 MATLAB 绘制 $x[n]$的图形，并验证(a)和(b)的结果。

2.20 (a) 写出图 P2.20 中各个序列的数学模型，形式为以 n 为变量的函数。

(b) 使用 MATLAB 生成并绘制(a)中得到函数，将结果与图 P2.20 进行比较验证。

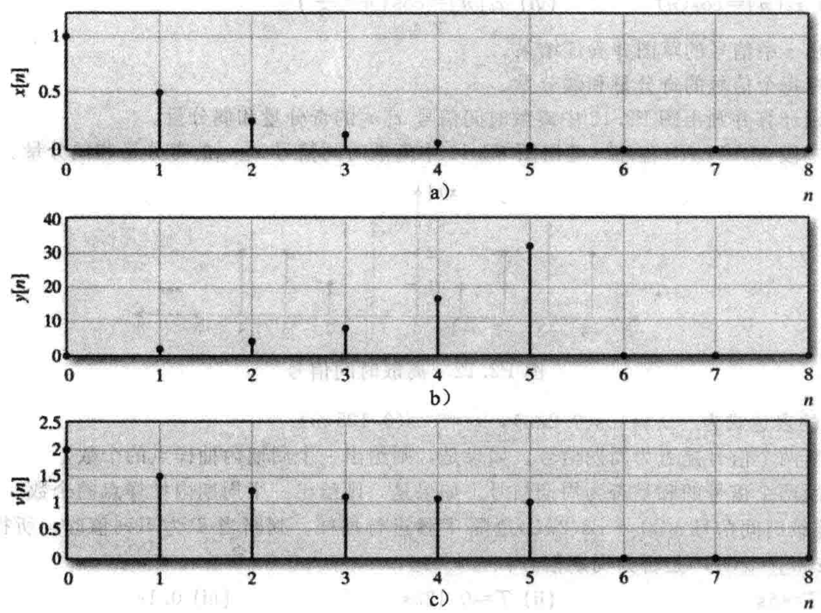


图 P2.20 离散时间序列

2.5 节

2.21 某系统的输出可以通过如下差分方程得到：

$$y[n]=5y[n-1]+y[n-2]+3x[n-1]-4x[n-2]$$

(a) 该系统是否为线性系统? 请验证答案。

(b) 该系统是否为非时变系统? 请验证答案。

(c) 该系统是否为因果系统? 请验证答案。

(d) 该系统是否为稳定系统? 请验证答案。

2.22 某 LTI 离散系统的脉冲响应为 $h[n]=(0.5)^{n-1}(u[n]-u[n-2])$，输入信号为 $x[n]=\delta[n]+\delta[n-1]$

(a) 该系统是否为因果系统? 请验证答案。

(b) 该系统是否为稳定系统? 请验证答案。

2.23 判断下列系统是否为因果系统。

(a) $y[n]=0.3x[n-2]+0.1x[n-1]$

(b) $y[n]=x[n+2]$

(c) $y[n]=-0.2x[n+1]+2x[n]+0.5x[n-1]+0.25x[n-2]$

(d) $y[n]=0.5x[n-1]+0.25x[n-2]+0.125x[n-3]+0.0625x[n+1]$

2.24 绘制如下差分方程描述的系统框图：

$x_1[n+1]=x_2[n]$

$x_2[n+1]=-0.5x_1[n]+0.125x_2[n]+u[n-1]$

$y[n]=2x_1[n]$

2.25 写出图 P2.25 所示系统的差分方程模型。

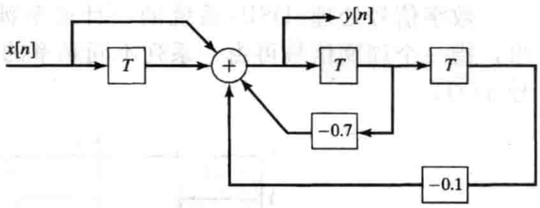


图 P2.25　离散时间系统

2.6 节

2.26 某线性非时变系统的输入信号为 $x[n]$，输出信号为 $y[n]$，脉冲响应为 $h[n]$。对于下列情况，给出系统的输出 $y[n]$。信号如图 P2.26 所示。

(a) $x[n]$为(a)信号，$h[n]$为(b)信号。

(b) $x[n]$为(b)信号，$h[n]$为(d)信号。

(c) $x[n]$为(a)信号，$h[n]$为(c)信号。

(d) $x[n]$为(d)信号，$h[n]$为(e)信号。

(e) $x[n]$为(b)信号，$h[n]$为(f)信号。

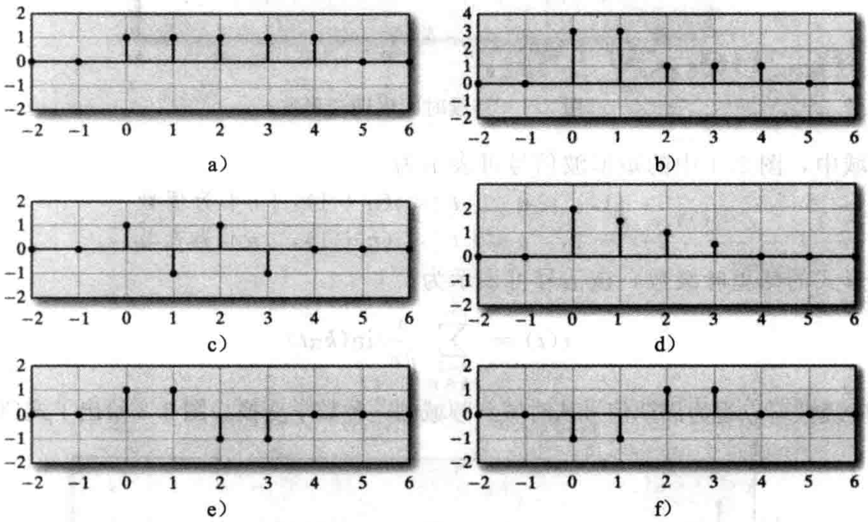


图 P2.26　离散时间信号

2.27 (a) 假设两个 LTI 离散系统级联，各自的脉冲响应相同，即 $h_1[n]=h_2[n]=(0.8)^n$，求出整个系统的响应。

(b) 重复过程(a)，其中 $h_1[n]=h_2[n]=\delta[n-3]$。

(c) 重复过程(a)，其中 $h_1[n]=h_2[n]=u[n]-u[n-2]$。

2.28 某 LTI 离散系统的脉冲响应为 $h[n]=3\delta[n]+\delta[n-1]+\delta[n-2]$，输入信号为 $x[n]=2\delta[n]+3\delta[n-1]+4\delta[n-2]+\delta[n-3]$，计算并写出输出信号 $y[n]$的数学表达式。

2.29 某 LTI 离散系统的脉冲响应为 $h[n]=(0.5)^{n-1}(u[n]-u[n-2])$，输入信号为 $x[n]=\delta[n]+\delta[n-1]$，计算并绘制输出信号 $y[n]$的图形。

2.30 某 LTI 离散系统的脉冲响应为 $h[n]=(0.5)^{n+1}(u[n]-u[n-2])$，输入信号为 $x[n]=\delta[n]+\delta[n-1]$

(a) 该系统是否为因果系统？

(b) 该系统是否为稳定系统？

第3章

频域概念

数字信号处理(DSP)系统的设计通常涉及频域。“频域”是由约瑟夫·傅里叶首先提出，指一个周期信号可由一系列不同频率的正弦信号表示。如图 3.1所示的时域矩形波信号 $x(t)$。

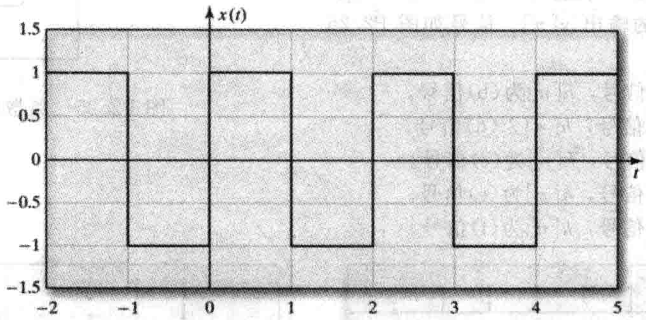


图 3.1　连续时间周期矩形波

在时域中，图 3.1 中的矩形波信号可表示为

$$x(t)=\begin{cases}1, & n<|t|<(n+1),\ |n|\ \text{为偶数}\\ -1, & n<|t|<(n+1),\ |n|\ \text{为奇数}\end{cases} \tag{3.1}$$

利用三角形式的傅里叶级数，该信号可表示为

$$x(t)=\sum_{\substack{k=1\\ k\text{为奇数}}}^{\infty}\frac{4}{k\pi}\sin(k\pi t) \tag{3.2}$$

傅里叶级数可以视为周期信号从时域到频域的一种数学变换。图 3.2 给出了式(3.2)中 $x(t)$

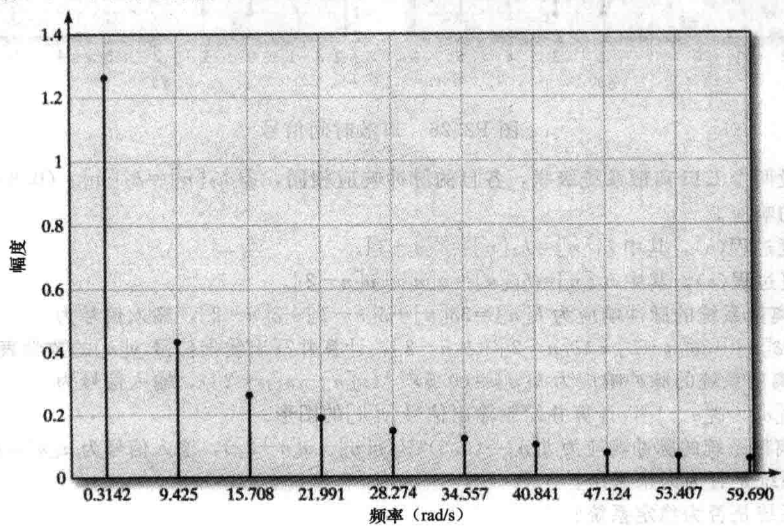


图 3.2　式(3.2)中傅里叶级数正弦分量的幅度-频率图

的傅里叶级数表示中正弦分量的幅度。对于傅里叶级数，时域信号是连续的，而傅里叶级数却只在离散的频率点是非零的。这两个域中的任意一个都可以是离散的或者连续的，这里总结了四种时-频关系，如表3.1所示。

表3.1 傅里叶分析方法的时-频特点

时间	频谱	傅里叶分析方法
连续	连续	傅里叶变换
连续	离散	傅里叶级数
离散	连续	离散时间傅里叶变换(DTFT)
离散	离散	离散傅里叶变换(DFT)

基于傅里叶分析可以得到其他两种变换：拉普拉斯变换是傅里叶变换的推广，z 变换是离散时间傅里叶变换(DTFT)的推广。

本章将介绍这些变换并为它们在DSP中的应用奠定基础。

频域概念滤波器的参数通常以其频率响应给出。因此，理解从时域到频域变换的数学概念非常重要，其可通过变换的方法或者通过等同于变换的映射函数来实现。

观察时域信号时，我们通常依靠示波器，它能产生实时的电压-时间图。为了在频域中"看到"信号，需要使用频谱分析仪，用于产生增益(电压)-频率图。通过频谱分析仪可以知道信号中含有哪些频率的正弦分量。从数学角度讲，也可以通过将信号从时域变换到频域来推断它的正弦频率成分。凭借傅里叶级数和傅里叶变换的机理，可将时域信号以其频率成分的形式重写。傅里叶级数应用于周期信号，傅里叶变换主要应用于非周期信号，下面将看到其也可以应用于周期信号。

3.1 正交函数和傅里叶级数

在很多情况下，分析一个线性系统的关键是合适地表示输入信号。周期信号常用的表示是傅里叶级数，傅里叶级数允许周期信号表示为一系列函数之和。这些函数通过系统的响应都可以简易得到。根据叠加性质，输出信号是各个函数各自响应的总和。傅里叶级数的表示正是基于对正交函数的使用，正交函数之和可以等于或者近似于一个信号。

函数的正交性基于向量正交性相同的思想。正交向量的分量用来表示多维向量，图3.3表示了一个三维空间的向量 v，其可由三个正交向量的分量表示，向量 v 可表示为

$$v = 3\hat{i} + 5\hat{j} + 2\hat{k}$$

式中，$\hat{i}$、$\hat{j}$和$\hat{k}$分别是 x 轴、y 轴和 z 轴方向的单位向量。因为$\hat{i}$、$\hat{j}$和$\hat{k}$相互正交，所以在彼此方向上的投影(即点积)存在 $\hat{i}\cdot\hat{j}=\hat{i}\cdot\hat{k}=\hat{j}\cdot\hat{k}=0$。用于表示信号的正交函数都具有该性质。

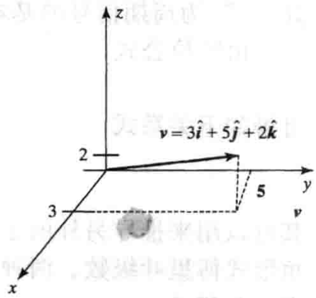


图3.3 三维空间的向量

如果希望用含有 n 个函数 $f_1(t)$，$f_2(t)$，$f_3(t)$，…，$f_n(t)$的函数集表示在一段时间内 $t_1\leqslant t\leqslant t_2$ 的信号，则要求在该特定时间段内，此 n 个函数满足正交，即，

$$\int_{t_1}^{t_2} f_i(t) f_j^*(t)\,\mathrm{d}t = \begin{cases} m_i, & i = j \\ 0, & i \neq j \end{cases} \tag{3.3}$$

其中，m_i 为常量，当 $f(t)$为复函数时，$f_j^*(t)$表示 $f_j(t)$的复共轭。若所有的 $m_i=1$，则称这些函数是标准正交的。信号 $x(t)$可用此正交函数集表示为

$$x(t) \approx c_1 f_1 + c_2 f_2 + c_3 f_3 + \cdots + c_n f_n;\quad t_1 \leqslant t \leqslant t_2 \tag{3.4}$$

在式(3.4)中没有用等号，因为用 $\sum_{k=1}^{n} c_k f_k(t)$ 表示 $x(t)$存在误差。计算系数 c_k 使得实际信

号与正交函数集表征信号之间的误差最小。

通常用于测量近似误差的一个准则是均方误差(MSE)。计算 $x(t)$ 与 $\sum_{k=1}^{n} c_k f_k(t)$ 之间的 MSE 为

$$\text{MSE} = \frac{1}{t_2 - t_1}\int_{t_1}^{t_2}\left[x(t) - \sum_{k=1}^{n} c_k f_k(t)\right]^2 \mathrm{d}t \tag{3.5}$$

由式(3.5)易知 MSE≥0，可用数学方法论证[1]当选择下列系数 c_k 时，由正交函数集表示的信号与 $x(t)$之间的 MSE 最小。

$$c_k = \frac{\int_{t_1}^{t_2} x(t) f_k^*(t)\mathrm{d}t}{\int_{t_1}^{t_2} f_k(t) f_k^*(t)\mathrm{d}t} \tag{3.6}$$

3.1.1 指数形式傅里叶级数

指数形式傅里叶级数常用来表示周期连续时间信号，其中 $\mathrm{e}^{\mathrm{j}\omega_0 t}$ 是正交函数集的基础。函数集 $\mathrm{e}^{\mathrm{j}k\omega_0 t}$ 在任意起点的 $T_0=(t_2-t_1)=2\pi/\omega_0$ 内都是正交的，即

$$\int_{t_1}^{t_1+T_0} \mathrm{e}^{\mathrm{j}k\omega_0 t}\mathrm{e}^{-\mathrm{j}m\omega_0 t}\mathrm{d}t = \begin{cases} T_0, & k=m \\ 0, & k\neq m \end{cases}$$

求解式(3.6)中的系数，可得

$$c_k = \frac{\int_{t_1}^{t_1+T_0} x(t)\mathrm{e}^{-\mathrm{j}k\omega_0 t}\mathrm{d}t}{\int_{t_1}^{t_1+T_0} \mathrm{e}^{\mathrm{j}k\omega_0 t}\mathrm{e}^{-\mathrm{j}k\omega_0 t}\mathrm{d}t} = \frac{\int_{t_1}^{t_1+T_0} x(t)\mathrm{e}^{-\mathrm{j}k\omega_0 t}\mathrm{d}t}{\int_{t_1}^{t_1+T_0} 1\mathrm{d}t} = \frac{1}{T_0}\int_{t_1}^{t_1+T_0} x(t)\mathrm{e}^{-\mathrm{j}k\omega_0 t}\mathrm{d}t \tag{3.7}$$

因此，利用指数形式傅里叶级数表示连续时间周期信号为

$$x(t) = \sum_{k=-\infty}^{\infty} c_k \mathrm{e}^{\mathrm{j}k\omega_0 t} \tag{3.8}$$

$$c_k = \frac{1}{T_0}\int_{t_1}^{t_1+T_0} x(t)\mathrm{e}^{-\mathrm{j}k\omega_0 t}\mathrm{d}t \tag{3.9}$$

其中 T_0 为周期信号的基本周期，$\omega_0=2\pi/T_0$。

由欧拉公式

$$\mathrm{e}^{\pm\mathrm{j}\theta} = \cos\theta \pm \mathrm{j}\sin\theta \tag{3.10}$$

可得如下关系式

$$\cos\theta = \frac{\mathrm{e}^{\mathrm{j}\theta}+\mathrm{e}^{-\mathrm{j}\theta}}{2}, \sin\theta = \frac{\mathrm{e}^{\mathrm{j}\theta}-\mathrm{e}^{-\mathrm{j}\theta}}{2\mathrm{j}} \tag{3.11}$$

其可以用来推导另外的正交函数集，包括 $\sin\omega_0 t$ 和 $\cos\omega_0 t$。这些三角函数通常用来组成三角形式傅里叶级数。两种形式的三角傅里叶级数以及它们与指数形式傅里叶级数的关系如表 3.2 所示。

表 3.2 傅里叶级数的表示形式

傅里叶级数表现形式	表达式
指数形式	$x(t) = \sum_{k=-\infty}^{\infty} c_k \mathrm{e}^{\mathrm{j}k\omega_0 t}$ $c_k = \frac{1}{T_0}\int_{T_0} x(t)\mathrm{e}^{-\mathrm{j}k\omega_0 t}\mathrm{d}t = \lvert c_k\rvert \mathrm{e}^{\mathrm{j}\theta_k}, c_{-k} = c_k^*$
组合三角形式	$x(t) = c_0 + \sum_{k=1}^{\infty} 2\lvert c_k\rvert\cos(k\omega_0 t+\theta_k)$

（续）

傅里叶级数表现形式	表达式
三角形式	$x(t)=a_0+\sum_{k=1}^{\infty}[a_k\cos(k\omega_0 t)+\mathrm{j}b_k\sin(k\omega_0 t)]$， $a_0=c_0=\frac{1}{T_0}\int_{T_0}x(t)\mathrm{d}t, 2c_k=a_k-\mathrm{j}b_k$ $a_k=\frac{2}{T_0}\int_{T_0}x(t)\cos(k\omega_0 t)\mathrm{d}t$ 及 $b_k=\frac{2}{T_0}\int_{T_0}x(t)\sin(k\omega_0 t)\mathrm{d}t$

实际上，通常使用有限的正交函数集表示一个信号，这种截短的指数形式的傅里叶级数可表示为

$$x(t)=\sum_{k=-N}^{N}c_k\mathrm{e}^{-\mathrm{j}k\omega_0 t} \tag{3.12}$$

式中，N 必须足够大以产生足够准确的傅里叶级数表示。

例 3-1 周期矩形脉冲的指数傅里叶级数表示

在此例中，画出了图 3.4 中矩形脉冲序列的频率分布(称为频谱)。该波形经常用于工程当中，如数字电脑时钟信号就是这种形式的周期矩形脉冲。

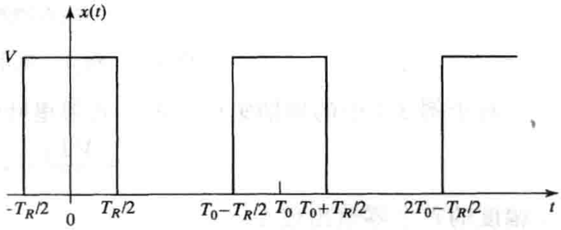


图 3.4 周期矩形脉冲

由式(3.9)得到该信号傅里叶级数的系数为

$$c_k=\frac{1}{T_0}\int_{-T_R/2}^{T_R/2}V\mathrm{e}^{-\mathrm{j}k\omega_0 t}\mathrm{d}t=\frac{V}{\mathrm{j}k\omega_0 T_0}[\mathrm{e}^{\mathrm{j}k\omega_0 T_R/2}-\mathrm{e}^{-\mathrm{j}k\omega_0 T_R/2}]=\frac{VT_R}{T_0}\left(\frac{1}{k\omega_0 T_R/2}\right)\frac{\mathrm{e}^{\mathrm{j}k\omega_0 T_R/2}-\mathrm{e}^{-\mathrm{j}k\omega_0 T_R/2}}{2\mathrm{j}}$$

运用欧拉关系式(3.10)，系数可写为

$$c_k=\frac{VT_R}{T_0}\left(\frac{\sin(k\omega_0 T_R/2)}{k\omega_0 T_R/2}\right)=\frac{VT_R}{T_0}\mathrm{sinc}(k\omega_0 T_R/2)$$

函数 $\mathrm{sinc}(x)$ 定义为 $\mathrm{sinc}(x)=\frac{\sin x}{x}$。(这个定义不是标准的，但是数学家通常使用该定义。在工程中，也定义 $\mathrm{sinc}(x)=\frac{\sin(\pi x)}{\pi x}$。)通过应用洛必达法则(L'Hôpital's)，可得 $\lim_{x\to 0}\mathrm{sinc}(x)=1$。当 $x\neq 0$ 时，$\mathrm{sinc}(x)$ 在 $\sin x$ 等于零的点处有零值。也就是在 $x=\pm\pi\pm 2\pi, \ldots$

周期矩形脉冲的傅里叶级数表示为

$$x(t)=\sum_{k=-\infty}^{\infty}\frac{VT_R}{T_0}\mathrm{sinc}(k\omega_0 T_R/2)\mathrm{e}^{\mathrm{j}k\omega_0 t}$$

其中，V 是幅度，T_R 是矩形脉冲的宽度。

图 3.5 显示了周期矩形脉冲信号的傅里叶级数表示中，各频率分量的幅度。

尽管指数形式的傅里叶级数由一系列正交的时域函数构成，但其提供了周期信号 $x(t)$ 中有关频率内容的关键信息。傅里叶系数通常是复数，可表示为极坐标形式

$$c_k=|c_k|\angle\theta_k$$

因此，傅里叶系数显示了信号在各个频率 $k\omega_0$ 处的幅度 $|c_k|$ 和相位角 θ_k。将幅度和相位角作为频率 $k\omega_0$ 的函数，可以分别绘出信号 $x(t)$ 对应的幅度-频率和相位-频率的分布图。 ◀

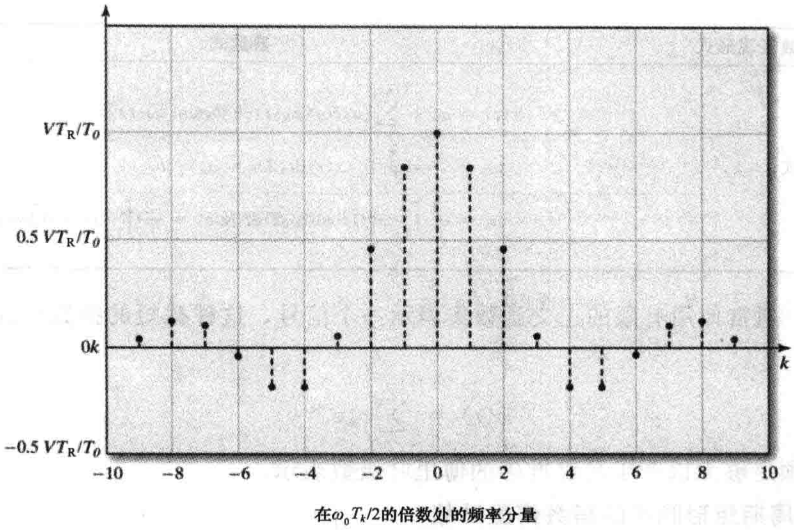


图 3.5　例 3-1 矩形脉冲的频率分量

对于例 3-1 中的周期矩形脉冲，其傅里叶级数为

$$c_k = \frac{VT_R}{T_0}\mathrm{sinc}(k\omega_0 T_R/2)$$

c_k 幅度的首个零值出现在

$$\sin(T_R\omega/2) = \sin(\pi)$$

如果使上式成立的 ω 值是离散频率集 $k\omega_0$ 的成员，则在 $\omega=2\pi/T_R$ 处，c_k 为零。通常情况下，ω 值不是 ω_0 的谐波频率。

图 3.5 给出了矩形脉冲序列的频谱图。该波形中所有 c_k 的值均为实数，因此当 c_k 为正数时，相位角为零；当 c_k 为负数时，相位角为 π，使得幅度和相位可在一幅图中显示。通常情况下，c_k 是复数，幅度谱和相位谱需分别绘出。在特定情况 $T_R=T_0/2$ 下，偶次谐波幅度为零，信号可由傅里叶级数的奇次谐波项表示。

频谱是信号与系统工程中的基本概念，信号与系统的许多分析和设计过程都是基于此概念。

3.1.2　离散傅里叶级数

利用离散时间正交基函数，离散时间周期信号可由离散傅里叶级数表示。离散时间基函数必须在展开区间 N_0 内相互正交，满足此正交要求的函数 $f[n]$需符合等式

$$\sum_{n=n_0}^{n_0+N_0-1} f_m[n]f_l^*[n] = \begin{cases} N_0, & l=m \\ 0, & l\neq m \end{cases}$$

通常用于离散傅里叶级数的基函数为复指数函数

$$f_m[n] = \mathrm{e}^{\mathrm{j}2\pi mn/N_0}$$

特别注意的是

$$f_{m+kN_0}[n] = \mathrm{e}^{\mathrm{j}2\pi(m+kN_0)n/N_0} = \mathrm{e}^{\mathrm{j}2\pi mn/N_0}\mathrm{e}^{\mathrm{j}2\pi kn} = f_m[n]$$

由此可见，该基函数只有 N_0 个不同的基信号。

离散傅里叶级数可表示为有限个基信号的加权和

$$x[n] = \sum_{k=k_0}^{k_0+N_0-1} X_k \mathrm{e}^{\mathrm{j}2\pi kn/N_0} \tag{3.13}$$

其中，X_k 是基信号的系数，利用基信号的正交性，X_k 可由以下公式给出：

$$X_k = \frac{1}{N_0}\sum_{n=n_0}^{n_0+N_0-1} x[n]\mathrm{e}^{-\mathrm{j}2\pi kn/N_0} \tag{3.14}$$

由式(3.13)和式(3.14)定义的离散傅里叶级数与3.4节所述的离散傅里叶变换(DFT)有着紧密联系。

3.1.3 傅里叶级数性质

狄利克雷(Dirichlet)条件。任何满足狄利克雷条件的单值周期函数 $x(t)$ 都可以展开为傅里叶级数。狄利克雷条件是[2]：

1. $x(t)$ 在一个周期内，只有有限个不连续点；
2. $x(t)$ 在一个周期内，只有有限个最大值和最小值；
3. $x(t)$ 有界。

第三条可以拓展为包含奇异函数，表述为[3]。

3a. $\int_{T_0} |x(t)|\mathrm{d}t < \infty$

在一个物理系统中出现的任何时间函数都满足以上条件。

对于满足狄利克雷条件1、2和3的 $x(t)$，以下表述均是正确的。

1. 如果在该点连续，且存在右导数和左导数，则无论左右导数是否相等，傅里叶级数都收敛于在该点处的函数值。$x(t)$ 在 $t=t_a$ 处的右导数定义为，当 t 从右侧趋近于 t_a 时的导数；左导数定义为，当 t 从左侧趋近于 t_a 时的导数。

2. 如果在某点处不连续，其傅里叶级数收敛于从右侧和从左侧趋近时极限值的平均值；即在每个 t_a 点处

$$\sum_{k=-\infty}^{\infty} C_k \mathrm{e}^{\mathrm{j}k\omega_0 t_a} = \frac{x(t_a^-)+x(t_a^+)}{2} \tag{3.15}$$

其中，$x(t_a^-)$ 是当 t 从左侧趋近于 t_a 时 $x(t)$ 的极限值，$x(t_a^+)$ 是从右侧趋近时的极限值。注意，式(3.15)对于连续点 t_a 同样适用。如果 $x(t)$ 有跳变，比如阶跃函数，傅里叶级数收敛于变化的中点(midpoint)。

3. 几乎任何周期为 T_0 的连续函数 $x(t)$ 都可以由指定精度的截断傅里叶级数来近似，级数由式(3.12)给出。

$$x_N(t) = \sum_{k=-N}^{N} C_k \mathrm{e}^{\mathrm{j}k\omega_0 t}$$

该性质适用于实际工程中可能遇到的任意连续周期函数。定义式(3.12)的截断级数近似误差为

$$e(t) = x(t) - x_N(t) \tag{3.16}$$

该性质表明，通过选择足够大的 N，可以将误差的绝对值 $|e(t)|$ 限定在任意精度范围。

4. 定义最小化均方误差(MSE)为

$$\mathrm{MSE} = \frac{1}{T_0}\int_{T_0} \mathrm{e}^2(t)\mathrm{d}t \tag{3.17}$$

也就是说，在式(3.12)表示的谐波级数中，没有任何其他系数的选择能够产生比式(3.17)中更小的均方误差。

5. 周期三角函数之和是其本身的傅里叶级数。

6. 除周期脉冲信号以外，当 k 充分大时，$x(t)$ 的第 k 次谐波的傅里叶系数的幅度至少以 $1/k$ 的速率降低。周期脉冲信号的频谱将在后续章节中利用傅里叶变换的方法进行讨论。

7. 周期函数之和的傅里叶级数等于函数傅里叶级数之和。当然，周期函数之和必须

是周期的；如果不是，则不存在傅里叶级数。

第3条性质非常重要，因为它表述出几乎任何连续周期函数都可以由任意精度的截断傅里叶级数进行近似表示。

图3.6显示了一个方波的误差。其中画出的是正半周方波波形、一次谐波、一次谐波和三次谐波之和，以及一至九次谐波中奇次谐波之和的波形。通过增加高次谐波，误差明显降低。

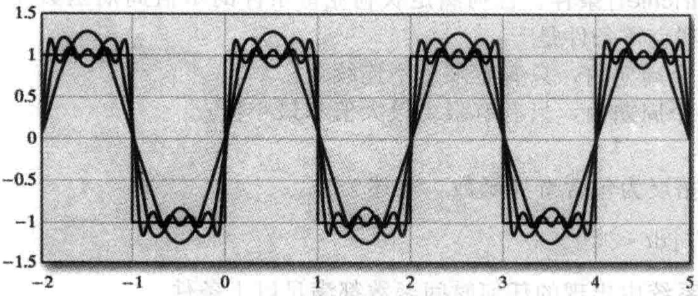


图3.6 周期方波的截断傅里叶级数近似

图3.6同时显示了吉伯斯现象。当叠加的项数增多时，级数波形中的起伏振荡随之变窄。然而，每个不连续点处的振荡的振幅不趋近于零，而是近似趋近于跳变高度的9%[2]。

3.2 傅里叶变换

傅里叶变换是在频域表示信号与系统的数学方法。工程师利用傅里叶变换简化信号与系统的数学分析，用数学方法解释物理现象。傅里叶变换在电气工程领域得到广泛应用，尤其在电子通信信号与系统的研究中更为突出。

3.2.1 傅里叶变换的定义

为了定义傅里叶变换，这里首先给出指数形式傅里叶级数，其中

[(3.8)]
$$x(t)=\sum_{k=-\infty}^{\infty}c_k\mathrm{e}^{\mathrm{j}k\omega_0 t}$$

以及

[(3.9)]
$$c_k=\frac{1}{T_0}\int_{T_0}x(t)\mathrm{e}^{-\mathrm{j}k\omega_0 t}\mathrm{d}t$$

此前讨论的周期函数，如图3.7a中信号，利用傅里叶级数的谐波成分表示。这里考虑增大周期信号的周期，如图3.7b及图3.7c中信号所示。在图3.7中，可以注意到随着周期的不断增大直至其周期变为无限长，可以使得图3.7c中的波形不再重复出现。

考虑式(3.9)中包含在被积函数中的指数函数，其随着 k 的增加，$k\omega_0$ 按 ω_0 的增量增加。定义频率增量为

$$\Delta\omega=(k+1)\omega_0-k\omega_0=\omega_0$$

由于 $\omega_0=2\pi/T_0$，频率增量随着波形周期 T_0 的增大而减少。在 T_0 趋近于无穷时，频率增量 $\Delta\omega$ 变为频率微分 $\mathrm{d}\omega$

$$\lim_{T_0\to\infty}\frac{2\pi}{T_0}=\mathrm{d}\omega$$

同样地，当 T_0 为无穷时，$k\omega_0=2\pi k/T_0$ 趋近于 $k\mathrm{d}\omega$。由于 k 在整数值中无限地变化，乘积 $k\mathrm{d}\omega$ 变成连续的频率变量 ω。因此，式(3.9)可写为

$$c_{k_\infty}=\lim_{T_0\to\infty}\frac{1}{2\pi}\frac{2\pi}{T_0}\int_{-T_0/2}^{T_0/2}x(t)\mathrm{e}^{-(\mathrm{j}k2\pi/T_0)t}\mathrm{d}t=\frac{1}{2\pi}\left[\int_{-\infty}^{\infty}x(t)\mathrm{e}^{-\mathrm{j}\omega t}\mathrm{d}t\right]\mathrm{d}\omega$$

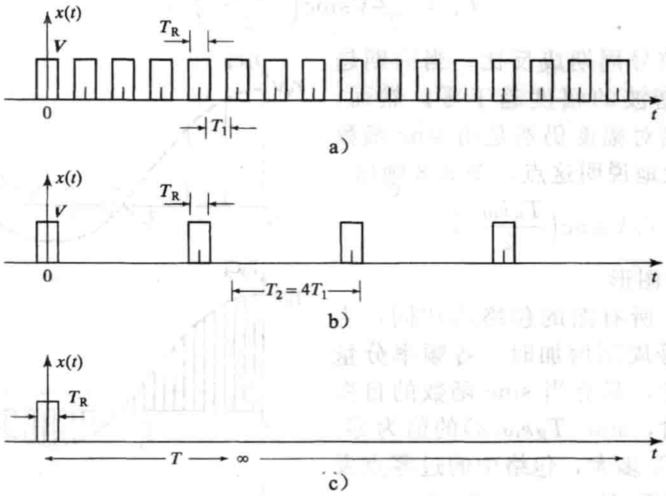


图 3.7 矩形脉冲

将上式括号中的函数定义为傅里叶变换，记为

$$\mathrm{F}\{x(t)\}=X(\omega)=\int_{-\infty}^{\infty}x(t)\mathrm{e}^{-\mathrm{j}\omega t}\mathrm{d}t \tag{3.18}$$

可以写为 $c_{k_\infty}=(1/2\pi)X(\omega)\mathrm{d}\omega$，因此，由式(3.8)可得

$$x(t)=\sum_{k=-\infty}^{\infty}\frac{1}{2\pi}X(\omega)\mathrm{d}\omega\mathrm{e}^{\mathrm{j}k\omega_0 t}=\frac{1}{2\pi}\sum_{k=-\infty}^{\infty}X(\omega)\mathrm{e}^{\mathrm{j}k\omega_0 t}\mathrm{d}\omega$$

将累加变为积分，$x(t)$的方程可写为

$$x(t)=\frac{1}{2\pi}\int_{-\infty}^{\infty}X(\omega)\mathrm{e}^{\mathrm{j}\omega t}\mathrm{d}\omega=\mathrm{F}^{-1}\{X(\omega)\} \tag{3.19}$$

如同此前的推导，其中我们利用了 $\lim\limits_{T_0\to\infty}k\omega_0=\omega$。

式(3.18)和式(3.19)分别定义了傅里叶变换与傅里叶反变换：

[(3.18)] $$\mathrm{F}\Big\{x(t)\Big\}=X(\omega)=\int_{-\infty}^{\infty}x(t)\mathrm{e}^{-\mathrm{j}\omega t}\mathrm{d}t$$

[(3.19)] $$\mathrm{F}^{-1}\Big\{X(\omega)\Big\}=x(t)=\frac{1}{2\pi}\int_{-\infty}^{\infty}X(\omega)\mathrm{e}^{\mathrm{j}\omega t}\mathrm{d}\omega$$

这些等式共同称为傅里叶变换对，它们的关系通常以数学符号记为

$$x(t)\overset{F}{\longleftrightarrow}X(\omega)$$

例 3-2 傅里叶变换的物理意义

为了理解式(3.18)和式(3.19)定义的傅里叶变换对物理意义，考虑图 3.7 中的周期矩形脉冲。在例 3-1 中，周期矩形脉冲的指数傅里叶级数表示为：

$$x(t)=\sum_{k=-\infty}^{\infty}C_k\mathrm{e}^{\mathrm{j}k\omega_0 t}=\sum_{k=-\infty}^{\infty}\frac{T_\mathrm{R}}{T_0}V\mathrm{sinc}\left(\frac{T_\mathrm{R}k\omega_0}{2}\right)\mathrm{e}^{\mathrm{j}k\omega_0 t} \tag{3.20}$$

各次谐波的幅度按照 sinc 函数变化，如图 3.8a 所示。

现在考虑信号周期从 $T_0=T_1$ 增加到 $T_0=T_2=4T_1$ 时的影响。由于 $\omega_0=2\pi/T_0$，随着周期的增加，频谱分量同时沿着频率轴越来越靠近，因此 $\Delta\omega=\omega_0$ 减小，如图 3.8b 所示。

当 $T_0\to\infty$时，各分量间隔变为无限小；换言之，$\Delta\omega$ 变成 $\mathrm{d}\omega$。因此，式(3.20)中的累加变为积分，离散的频谱变成一条连续曲线。信号频谱的各次谐波的复系数为

$$C_k = \frac{T_R}{T_0} V \operatorname{sinc}\left(\frac{T_R k\omega_0}{2}\right)$$

各系数的幅度与信号周期成反比。当周期趋于无穷时，各次谐波的幅度趋于零。然而，信号谐波之间的相对幅度仍然是由 sinc 函数决定。为了更清楚地说明这点，图 3.8 画出

$$T_0 C_k = T_R V \operatorname{sinc}\left(\frac{T_R k\omega_0}{2}\right)$$

随 $\omega(\omega \geqslant 0)$变化的图形。

在图 3.8 中，所有图的包络均相同，主要差别就是当信号周期增加时，各频率分量越来越密集。注意，只有当 sinc 函数的自变量是 π 的整数倍时，$\operatorname{sinc}(T_R k\omega/2)$的值为零。因此，无论周期 T_0 多大，包络中的过零点发生在频率 $\omega=2\pi n/T_R$ 处，$n=1, 2, 3, \cdots$ ◀

图 3.8c 所示的频谱是幅度为 V，持续时间为 T_R 的单个矩形脉冲(也可以认为是一个周期为无限长的周期矩形脉冲)傅里叶变换的图形。利用式(3.18)可得到傅里叶变换的解析表达式，矩形脉冲可用数学形式描述为两个阶跃函数之和：

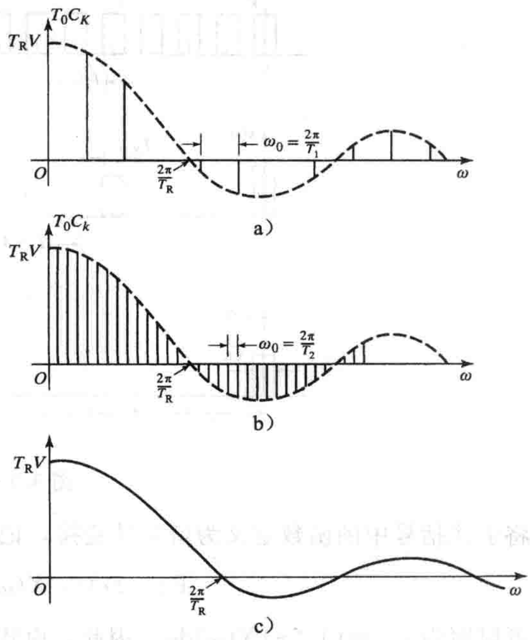


图 3.8　矩形脉冲的频谱

$$x(t) = Vu(t+T_R/2) - Vu(t-T_R/2) = V[u(t+T_R/2) - u(t-T_R/2)]$$

由于电子通信系统中经常遇到矩形脉冲，时常将其定义为一个函数：

$$\operatorname{rect}(t/T_R) \equiv [u(t+T_R/2) - u(t-T_R/2)]$$

为了化简式(3.18)中的积分，在$-T_R/2<t<+T_R/2$ 时间段内，$x(t)$的值为 V，其他时间内 $x(t)$均为零。因此，

$$X(\omega) = \int_{-T_R/2}^{+T_R/2} V\mathrm{e}^{-\mathrm{j}\omega t}\,\mathrm{d}t = V\left[\left|\frac{\mathrm{e}^{-\mathrm{j}\omega t}}{-\mathrm{j}\omega}\right|_{-T_R/2}^{+T_R/2}\right] = V\left[\frac{\mathrm{e}^{-\mathrm{j}T_R\omega/2} - \mathrm{e}^{+\mathrm{j}T_R\omega/2}}{-\mathrm{j}\omega}\right]$$

$$= \frac{T_R V}{\omega T_R/2}\left[\frac{\mathrm{e}^{\mathrm{j}T_R\omega/2} - \mathrm{e}^{-\mathrm{j}T_R\omega/2}}{\mathrm{j}2}\right] = T_R V\left[\frac{\sin(T_R\omega/2)}{T_R\omega/2}\right] = T_R V \operatorname{sinc}(T_R\omega/2)$$

此时，我们得到非周期矩形脉冲信号的傅里叶变换：

$$\mathrm{F}\{V\operatorname{rect}(t/T_R)\} = VT_R \operatorname{sinc}(T_R\omega/2)$$

写成变换对形式，

$$V\operatorname{rect}(t/T_R) \overset{\mathrm{F}}{\longleftrightarrow} VT_R \operatorname{sinc}(T_R\omega/2) \tag{3.21}$$

值得注意的是，非周期矩形脉冲信号的傅里叶变换与例 3-1 中周期矩形脉冲的傅里叶级数表示有相同形式的包络。

尽管我们还未讨论某些信号不存在傅里叶变换，但不影响式(3.21)所示变换对的正确性。

存在傅里叶变换的条件　傅里叶变换存在的充分条件与傅里叶级数存在条件相似——狄利克雷条件：

1. 在任意有限长区间内，
   a. $x(t)$有界；
   b. $x(t)$有有限个最大值和最小值；
   c. $x(t)$有有限个不连续点。

2. $x(t)$绝对可积，即，

$$\int_{-\infty}^{\infty} |x(t)|\,\mathrm{d}t < \infty \tag{3.22}$$

以上条件为充分条件并非必要条件。如果这些是必要条件，那么利用傅里叶变换将无法分析许多有用的信号。

在实际应用中，对于工程师想要分析的信号或者函数，可以利用这样的规则：如果我们可以画出 $x(t)$的时域波形，那么该信号就存在傅里叶变换。实际上，对于一些无法准确画出波形的信号(如冲激函数)，其也存在傅里叶变换。总之，对于所有物理可实现的信号都存在傅里叶变换。

函数的傅里叶变换 实际上，冲激函数为一些更为重要的变换对提供了基础。考虑信号

$$x(t) = A\delta(t-t_0)$$

其表示权值为 A 的冲激函数，只在 $t=t_0$ 时刻有非零值。此信号的傅里叶变换为

$$X(\omega) = \mathrm{F}\{A\delta(t-t_0)\} = \int_{-\infty}^{\infty} A\delta(t-t_0)\mathrm{e}^{-\mathrm{j}\omega t}\,\mathrm{d}t$$

回顾式(2.41)所描述的冲激函数的筛选性质，即

$$\int_{-\infty}^{\infty} x(t)\delta(t-t_0)\,\mathrm{d}t = x(t_0) \tag{3.23}$$

$x(t)$在 $t=t_0$ 处连续。利用冲激函数的此性质计算冲激函数的傅里叶变换，可得

$$X(\omega) = \mathrm{F}\{A\delta(t-t_0)\} = A\mathrm{e}^{-\mathrm{j}\omega t_0} = A\angle -\omega t_0 \tag{3.24}$$

可以看出，任意频率处 $X(\omega)$幅度为常数 A，相位角为频率的线性函数。

前面讨论过冲激函数的特例，在 $t=0$ 处的单位冲激函数，

$$x(t) = \delta(t)$$

由式(3.21)中的 $A=1$，$t_0=0$ 可以得到

$$\delta(t) \overset{\mathrm{F}}{\longleftrightarrow} 1 \tag{3.25}$$

当我们处理冲激函数时，也考虑频域中冲激函数，即

$$X(\omega) = \delta(\omega-\omega_0)$$

其具有与 $\delta(t-t_0)$相同的性质，

$$\delta(\omega-\omega_0) = \begin{cases} \to\infty, & \omega=\omega_0 \\ 0, & \omega\neq\omega_0 \end{cases}$$

$$\int_{-\infty}^{\infty} \delta(\omega-\omega_0)\,\mathrm{d}\omega = 1$$

$$X(\omega)\delta(\omega-\omega_0) = X(\omega_0)\delta(\omega-\omega_0)$$

$$\int_{-\infty}^{\infty} X(\omega)\delta(\omega-\omega_0)\,\mathrm{d}\omega = X(\omega_0)$$

由式(3.19)可得频域冲激函数的傅里叶反变换：

$$x(t) = \mathrm{F}^{-1}\{X(\omega)\} = \mathrm{F}^{-1}\{\delta(\omega-\omega_0)\} = \frac{1}{2\pi}\int_{-\infty}^{\infty} \delta(\omega-\omega_0)\mathrm{e}^{\mathrm{j}\omega t}\,\mathrm{d}\omega$$

应用冲激函数筛选性质可得

$$x(t) = \mathrm{F}^{-1}\{\delta(\omega-\omega_0)\} = \frac{1}{2\pi}\mathrm{e}^{\mathrm{j}\omega_0 t}$$

此傅里叶变换对记为

$$\mathrm{e}^{\mathrm{j}\omega_0 t} \overset{\mathrm{F}}{\longleftrightarrow} 2\pi\delta(\omega-\omega_0) \tag{3.26}$$

表 3.3 列出了部分连续时间函数的傅里叶变换。

表 3.3 傅里叶变换对

时域信号	傅里叶变换
$x(t)$	$\int_{-\infty}^{\infty} x(t)\mathrm{e}^{-\mathrm{j}\omega t}\mathrm{d}t$
$\frac{1}{2\pi}\int_{-\infty}^{\infty} X(\omega)\mathrm{e}^{\mathrm{j}\omega t}\mathrm{d}\omega$	$X(\omega)$
$\delta(t)$	1
$A\delta(t-t_0)$	$A\mathrm{e}^{-\mathrm{j}\omega t_0}$
$u(t)$	$\pi\delta(\omega)+\frac{1}{\mathrm{j}\omega}2\pi A\delta(\omega)$
$\mathrm{sgn}(t)$	$\frac{2}{\mathrm{j}\omega}$
$\mathrm{e}^{\mathrm{j}\omega_0 t}$	$2\pi\delta(\omega-\omega_0)$
$\cos(\omega_0 t)$	$\pi[\delta(\omega-\omega_0)+\delta(\omega+\omega_0)]$
$\sin(\omega_0 t)$	$-\mathrm{j}\pi[\delta(\omega-\omega_0)-\delta(\omega+\omega_0)]$
矩形脉冲$=u\left(t+\frac{T_R}{2}\right)-u\left(t-\frac{T_R}{2}\right)$	$T_R\mathrm{sinc}\left(\frac{T_R\omega}{2}\right)$
$\cos(\omega_0 t)u(t)$	$\frac{\pi}{2}[\delta(\omega-\omega_0)+\delta(\omega+\omega_0)]-\frac{\mathrm{j}\omega}{\omega^2-\omega_0^2}$
$\sin(\omega_0 t)u(t)$	$\frac{-\mathrm{j}\pi}{2}[\delta(\omega-\omega_0)-\delta(\omega+\omega_0)]-\frac{\omega}{\omega^2-\omega_0^2}$
$\cos(\omega_0 t)[u(t+T_R/2)-u(t-T_R/2)]$	$\frac{T_R}{2}\left[\mathrm{sinc}\left(\frac{T_R(\omega-\omega_0)}{2}\right)+\mathrm{sinc}\left(\frac{T_R(\omega+\omega_0)}{2}\right)\right]$
$\frac{\beta}{\pi}\mathrm{sinc}(\beta t)$	$u(\omega+\beta)-u(\omega-\beta)$
三角形脉冲$=\begin{cases}1-\frac{\lvert t\rvert}{T_W}, & \lvert t\rvert<T_W\\ 0, & \lvert t\rvert\geqslant T_W\end{cases}$	$T_W\mathrm{sinc}^2\left(\frac{T_W\omega}{2}\right)$
$\mathrm{e}^{-at}u(t)$，$\mathrm{Re}\{a\}>0$	$\frac{1}{a+\mathrm{j}\omega}$
$\sum_{n=-\infty}^{\infty} x(t-nT_0)$	$\sum_{k=-\infty}^{\infty}\omega_0 X(k\omega_0)\delta(\omega-k\omega_0),\omega_0=\frac{2\pi}{T_0}$
$\delta_T(t)=\sum_{n=-\infty}^{\infty}\delta(t-nT)$	$\omega_s\sum_{k=-\infty}^{\infty}\delta(\omega-k\omega_s),\omega_s=\frac{2\pi}{T}$

下面讨论傅里叶变换的一些性质。

3.2.2 傅里叶变换性质

傅里叶变换的某些性质可以极大地简化其在信号与系统分析中的应用。表 3.4 列出了常用的傅里叶变换性质，本节将对其中的一些性质进行详细说明。

表 3.4 傅里叶变换性质

操作	时域函数	傅里叶变换
线性	$ax_1(t)+bx_2(t)$	$aX_1(\omega)+bX_2(\omega)$
时移	$x(t-t_0)$	$X(\omega)\mathrm{e}^{-\mathrm{j}\omega t_0}$
时域展缩	$x(at)$	$\frac{1}{\lvert a\rvert}X\left(\frac{\omega}{a}\right)$
时域变换	$x(at-t_0)$	$\frac{1}{\lvert a\rvert}X\left(\frac{\omega}{a}\right)\mathrm{e}^{-\mathrm{j}\omega t_0/a}$
对偶性	$X(t)$	$2\pi x(-\omega)$
频移	$x(t)\mathrm{e}^{\mathrm{j}\omega_0 t}$	$X(\omega-\omega_0)$

(续)

操作	时域函数	傅里叶变换
卷积	$x_1(t) * x_2(t)$	$X_1(\omega)X_2(\omega)$
	$x_1(t)x_2(t)$	$\frac{1}{2\pi}[X_1(\omega) * X_2(\omega)]$
时域微分	$\frac{d^n x(t)}{dt^n}$	$(j\omega)^n X(\omega)$
频域微分	$(-jt)^n x(t)$	$\frac{d^n X(\omega)}{d\omega^n}$
时域积分	$\int_{-\infty}^{t} x(\tau)d\tau$	$\frac{1}{j\omega}X(\omega)+\pi X(0)\delta(\omega)$

这里仅利用应用举例来说明某些性质，并没有进一步的证明。希望学习这些性质的数学证明的读者，可阅读参考文献[1～3]。

例 3-3 线性和时移

考虑频率为 $\omega=100\pi$，在传输过程中延时 2.5ms 的时移余弦波：

$$x(t) = \cos[100\pi(t-2.5\times10^{-3})]$$

该信号可以看作是相移了π/4弧度的相移余弦波：

$$x(t) = \cos(100\pi t - 0.25\pi)$$

利用线性和时移性质，得到该延时余弦波的傅里叶变换：

$$F\{x(t)\} = X(\omega) = F\{\cos(100\pi t)\}e^{-j0.0025\omega} = \pi[\delta(\omega-100\pi)+\delta(\omega+100\pi)]e^{-j0.0025\omega}$$
$$= \pi[\delta(\omega-100\pi)e^{-j\pi/4}+\delta(\omega+100\pi)e^{j\pi/4}]$$

由于频谱中除了在 $\omega=100\pi$ 和 $\omega=-100\pi$ 处之外均为零幅度，因此与频率相关的复指数 $e^{-j0.0025\omega}$归结到最后等式中的两个固定的频率。回顾

$$X(\omega)\delta(\omega-\omega_0) = X(\omega_0)\delta(\omega-\omega_0)$$

频谱中清晰地显示了余弦波传输过程中的 2.5ms 时延所导致的$-\pi/4$弧度的相移。◀

例 3-4 时域卷积

在分析信号与系统的相互作用时，工程师们常利用卷积性质。第 2 章讨论了线性非时变系统对输入信号的响应，在图 3.9a 中给出了一个线性系统的框图。如果该系统在输入冲激函数时的输出响应记为 $h(t)$，则 $h(t)$ 称为该系统的冲激响应。系统对于任意输入信号的输出响应可以由冲激信号 $h(t)$和输入信号 $x(t)$的卷积确定。

$$y(t) = x(t) * h(t) = \int_{-\infty}^{\infty} x(\tau)h(t-\tau)d\tau$$

利用傅里叶变换的卷积性质，输出信号的频谱可由 $Y(\omega)=X(\omega)H(\omega)$ 得到。

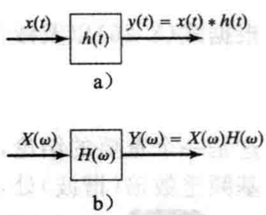


图 3.9 线性系统

其中，$h(t)\overset{F}{\longleftrightarrow}H(\omega)$，$x(t)\overset{F}{\longleftrightarrow}X(\omega)$，及 $y(t)\overset{F}{\longleftrightarrow}Y(\omega)$。函数 $H(\omega)$是系统的传输函数，在图 3.9b 中显示了频域中的信号/系统关系框图。◀

例 3-5 频移性质

为了说明傅里叶变换的频移性质，现推导两个连续时间正弦信号乘积的频谱：

$$v_3(t) = v_1(t)v_2(t)$$

其中，$v_1(t)=5\cos(2000\pi t)$，$v_2(t)=3\cos(300000\pi t)$。

在表 3.3 中，$F\{\cos(\omega_0 t)\}=\pi[\delta(\omega-\omega_0)+\delta(\omega+\omega_0)]$，根据表 3.4 中的频移性质，则有 $F\{x(t)e^{j\omega_0 t}\}=X(\omega-\omega_0)$。

对于 $v_1(t)$，利用欧拉关系式(3.11)将信号表示为两个复指数函数之和，

$$5\cos(2000\pi t) = 2.5(\mathrm{e}^{\mathrm{j}2000\pi t} + \mathrm{e}^{-\mathrm{j}2000\pi t})$$

因此乘积写为

$$v_3(t) = 7.5(\mathrm{e}^{\mathrm{j}2000\pi t}\cos(300000\pi t) + \mathrm{e}^{-\mathrm{j}2000\pi t}\cos(300000\pi t))$$

利用频移性质得到 $v_3(t)$ 的傅里叶变换。

$$\begin{aligned} v_3(\omega) = & 7.5\pi[\delta(\omega - 300000\pi - 2000\pi) + \delta(\omega - 300000\pi + 2000\pi) \\ & + \delta(\omega + 300000\pi - 2000\pi) + \delta(\omega + 300000\pi + 2000\pi)] \\ = & 7.5\pi[\delta(\omega - 298000\pi) + \delta(\omega - 302000\pi) + \delta(\omega + 298000\pi) + \delta(\omega + 302000\pi)] \end{aligned}$$

由此可见，$v_3(t)$ 的频谱由频域中的 4 个冲激组成，分别位于 $v_1(t)$ 和 $v_2(t)$ 正负频率的和与差处。每个冲激的强度为 7.5π。

频域的冲激可以重组为

$$V_3(\omega) = 7.5\{\pi[\delta(\omega - 298000\pi) + \delta(\omega + 298000\pi)] + \pi[\delta(\omega - 302000\pi) + \delta(\omega + 302000\pi)]\}$$

从上式的构成可以看出，该式表示了两个正弦函数傅里叶变换之和。利用线性性质可得到其对应的时域函数 $v_3(t)$。

$$v_3(t) = 7.5[\cos(302000\pi t) + \cos(298000\pi t)]$$

此例说明了无线通信中的调制概念，其利用载波信号去调制待传输的信号。

3.2.3 周期信号的傅里叶变换

由 3.1 节可知，时域的周期函数可以由傅里叶级数表示，即

[(3.8)]
$$x(t) = \sum_{k=-\infty}^{\infty} c_k \mathrm{e}^{\mathrm{j}k\omega_0 t}$$

式中，

[(3.9)]
$$c_k = \frac{1}{T_0}\int_{T_0} x(t)\mathrm{e}^{-\mathrm{j}k\omega_0 t}\mathrm{d}t$$

现在我们将推导出周期信号的傅里叶变换。

利用(3.15)以及式(3.8)的傅里叶变换，可得

$$X(\omega) = \int_{-\infty}^{\infty}\left[\sum_{k=-\infty}^{\infty} c_k \mathrm{e}^{\mathrm{j}k\omega_0 t}\right]\mathrm{e}^{-\mathrm{j}\omega t}\mathrm{d}t = \sum_{k=-\infty}^{\infty} c_k \int_{-\infty}^{\infty}(\mathrm{e}^{\mathrm{j}k\omega_0 t})\mathrm{e}^{-\mathrm{j}\omega t}\mathrm{d}t$$

根据式(3.26)以及线性性质，可得

$$\sum_{k=-\infty}^{\infty} c_k \mathrm{e}^{\mathrm{j}k\omega_0 t} \overset{\mathrm{F}}{\longleftrightarrow} 2\pi\sum_{k=-\infty}^{\infty} c_k\delta(\omega - k\omega_0) \tag{3.27}$$

这是一个重要的结论，周期信号的频谱是频域冲激串函数，这些冲激串位于此周期信号的基频整数倍(谐波)处，每个冲激函数的强度是其傅里叶级数的复系数乘以 2π。 ◀

例 3-6 周期冲激信号的频谱

计算周期冲激串的频谱。周期冲激串信号的数学描述为

$$x(t) = \sum_{n=-\infty}^{\infty}\delta(t - nT_0)$$

该周期信号的傅里叶系数为

$$c_k = \frac{1}{T_0}\int_{T_0}\delta(t)\mathrm{e}^{-\mathrm{j}k\omega_0 t}\mathrm{d}t = \frac{1}{T_0}\int_{T_0}\delta(t)\mathrm{d}t = \frac{1}{T_0}$$

因此，

$$X(\omega) = 2\pi\sum_{k=-\infty}^{\infty}\frac{1}{T_0}\delta(\omega - k\omega_0) = \omega_0\sum_{k=-\infty}^{\infty}\delta(\omega - k\omega_0)$$

上式结果为频域冲激串。注意，周期冲激串函数 $x(t)$ 的傅里叶变换 $X(\omega)$ 是一个周期为 ω_0

的周期冲激函数。◀

3.3　离散时间傅里叶变换

一旦通过抽样将连续时间信号转换为数字序列，就可以利用变换来分析信号以及数据处理算法的特性。本节定义了离散时间傅里叶变换(DTFT)。离散时间傅里叶变换应用于离散时间信号，等同于连续时间信号与系统中的傅里叶变换。考虑图 3.10 中冲激调制器的输出。

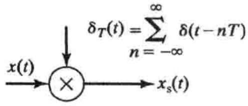


图 3.10　冲激抽样

利用冲激函数的筛选性质，冲激调制器的输出为

$$x_s(t)=\sum_{n=-\infty}^{\infty}x(nT)\delta(t-nT) \tag{3.28}$$

时移冲激函数的傅里叶变换为

$$\delta(t-t_0)\overset{F}{\longleftrightarrow}e^{-j\omega t_0}$$

由于 $x(nT)$不是连续时间的函数，因此利用傅里叶变换的线性性质可得，

$$X_s(\omega)=\sum_{n=-\infty}^{\infty}x(nT)e^{-jn\omega T}$$

如果将该等式中的 $x(nT)$用符号 $x[n]$表示，并用 Ω 表示 ωT，则定义离散时间傅里叶变换的表达式为：

$$X(\Omega)=\mathrm{DTF}\{x[n]\}=\sum_{n=-\infty}^{\infty}x[n]e^{-jn\Omega} \tag{3.29}$$

等式中的 DTF{·}代表离散时间傅里叶变换。DTFT 可以通过图 3.10 中信号抽样模型的傅里叶变换而得到。式(3.29)也可以通过对傅里叶变换(3.18)进行数学积分和矩形求和得到。

反 DTFT 定义为

$$x[n]=\mathrm{DTF}^{-1}\{X(\Omega)\}=\frac{1}{2\pi}\int_{\Omega_1}^{\Omega_1+2\pi}X(\Omega)e^{jn\Omega}d(\Omega)=\frac{1}{2\pi}\int_{2\pi}X(\Omega)e^{jn\Omega}d(\Omega) \tag{3.30}$$

其中，积分号正下方的值 2π 意味着 Ω_1 可为任意值，被积函数是周期为 2π 的周期函数。反积分可以直接由傅里叶变换的反积分(3.19)推导而得到。离散时间傅里叶变换对可表示为

$$x[n]\overset{\mathrm{DTF}}{\longleftrightarrow}X(\Omega) \tag{3.31}$$

需要特别注意的是，$x[n]$是离散变量 n 的函数，由于 $\Omega=\omega T$，式(3.29)中的 $X(\Omega)$是连续变量 Ω 的函数。因此，$X(\Omega)$是频率的连续函数，而 $x[n]$是时间的离散函数。

通常，离散序列 $x[n]$通过对连续时间信号 $x(t)$抽样得到，或者该序列被解释为连续时间信号的样本。容易知道，离散频率 $\Omega=\omega T$ 是频率 ω 缩放后的形式。

下面我们分析 DTFT 的两个例子。

例 3-7　离散时间指数信号的离散时间傅里叶变换(DTFT)

计算函数 $x[n]=a^n u[n]$的 DTFT，其中

$$a^n u[n]=\begin{cases}a^n, & n\geqslant 0\\ 0, & n<0\end{cases}$$

回顾 2.4 节，$x[n]$本质上是一个指数函数。由式(3.29)

$$X(\Omega)=\sum_{n=-\infty}^{\infty}x[n]e^{-jn\Omega}=\sum_{n=0}^{\infty}a^n e^{-jn\Omega}=1+ae^{-j\Omega}+a^2e^{-j2\Omega}+\cdots=1+ae^{-j\Omega}+(ae^{-j\Omega})^2+\cdots$$

为了得到闭合形式的 $X(\Omega)$，利用几何级数

$$\sum_{n=0}^{\infty} b^n = 1 + b + b^2 + \cdots = \frac{1}{1-b}; \quad |b| < 1$$

此级数闭合形式的正确性可由长除法验证：

$$\frac{1}{1-b} = 1-b\overline{)1} \quad \text{商: } 1+b+b^2+b^3+\cdots$$

$$\begin{array}{l} \underline{1-b} \\ \quad b \\ \quad \underline{b-b^2} \\ \qquad b^2 \\ \qquad \underline{b^2-b^3} \\ \qquad\quad b^3 \\ \qquad\quad \cdots \end{array}$$

在 $X(\Omega)$中，令 $a\mathrm{e}^{-\mathrm{j}\Omega}=b$，变换的结果为

$$X(\Omega) = \frac{1}{1-a\mathrm{e}^{-\mathrm{j}\Omega}}; \quad |a\mathrm{e}^{-\mathrm{j}\Omega}| < 1$$

由于$|\mathrm{e}^{-\mathrm{j}\Omega}|=|\cos\Omega-\mathrm{j}\sin\Omega|=1$，因此当$|a|<1$时，此变换存在；于是存在下列变换对

$$a^n u[n] \overset{\text{DTF}}{\longleftrightarrow} \frac{1}{1-a\mathrm{e}^{-\mathrm{j}\Omega}}, \quad |a| < 1$$

无论 a 为实数还是复数，该变换均有效。当$|a|>1$时，$a^n u[n]$的 DTFT 不存在。 ◀

表 3.5 列出了部分离散时间函数的 DTFT。

表 3.5 离散时间傅里叶变换对

离散时间序列	DTFT
$\delta[n]$	1
$\delta[n-m]$	$\mathrm{e}^{-\mathrm{j}m\Omega}$
$u[n]$	$\dfrac{\mathrm{e}^{\mathrm{j}\Omega}}{\mathrm{e}^{\mathrm{j}\Omega}-1}+\pi\sum\limits_{k=-\infty}^{\infty}\delta(\Omega-2\pi k)$
$u[n]-u[n-n_0]$	$\dfrac{\sin(n_0\Omega/2)}{\sin(\Omega/2)}\mathrm{e}^{-\Omega(n_0-1)/2}$
$a^n u[n]$, $\lvert a\rvert<1$	$\dfrac{\mathrm{e}^{\mathrm{j}\Omega}}{\mathrm{e}^{\mathrm{j}\Omega}-a}$
$-a^n u[-n-1]$, $\lvert a\rvert>1$	$\dfrac{\mathrm{e}^{\mathrm{j}\Omega}}{\mathrm{e}^{\mathrm{j}\Omega}-a}$
$na^n u[n]$, $\lvert a\rvert<1$	$\dfrac{a\mathrm{e}^{\mathrm{j}\Omega}}{(\mathrm{e}^{\mathrm{j}\Omega}-a)^2}$
任意 n 处的值均为 1	$2\pi\sum\limits_{k=-\infty}^{\infty}\delta(\Omega-2\pi k)$

3.3.1 DTFT 的性质

本节给出 DTFT 的性质，这些性质都是基于离散时间傅里叶变换的定义式，

$$X(\Omega) = \sum_{n=-\infty}^{\infty} x[n]\mathrm{e}^{-\mathrm{j}n\Omega} \quad [(3.29)]$$

及

$$x[n] = \frac{1}{2\pi}\int_{2\pi} X(\Omega)\mathrm{e}^{\mathrm{j}n\Omega}\,\mathrm{d}\Omega \quad [(3.30)]$$

这里首先看看傅里叶变换与 DTFT 之间的关系。考虑以抽样周期为 T 秒的周期冲激串对信号 $x(t)$进行抽样所得信号 $x_s(t)$的 DTFT。

$$F[x_s(t)]_{t=nT} = F\left[\sum_{n=-\infty}^{\infty} x(nT)\delta(t-nT)\right]_{\Omega T=\omega} = \text{DTF}\{x(nT) = x[n]\} \qquad (3.32)$$

由此可以看出，可认为 DTFT 是带有权重的冲激函数串的傅里叶变换。3.2 节给出的傅里叶变换的性质可直接应用到 DTFT，只是按照式(3.32)的定义对某些变量进行修改。表 3.6列出了 DTFT 性质。

表 3.6 DTFT 性质

操作	离散序列	DTFT
线性	$ax_1[n]+bx_2[n]$	$aX_1(\Omega)+bX_2(\Omega)$
时移	$x[n-n_0]$	$X(\Omega)e^{-j\Omega n_0}$
频移	$x[n]e^{j\Omega_0 n}$	$X(\Omega-\Omega_0)$
频域微分	$nx[n]$	$j\dfrac{dX(\Omega)}{d\Omega}$
周期性	$x[n]$	$X(\Omega+2\pi k)=X(\Omega)$
共轭对称	$x^*[n]$	$X^*(-\Omega)$
时域翻转	$x[-n]$	$X(-\Omega)$
时域卷积	$x_1[n]*x_2[n]$	$X_1(\Omega)X_2(\Omega)$
频域卷积(时域乘积)	$x_1[n]x_2[n]$	$\dfrac{1}{2\pi}\int_{2\pi}X_1(\lambda)X_2(\Omega-\lambda)d\lambda$
帕赛瓦尔定理	$\sum_{n=-\infty}^{\infty}\|x[n]\|^2$	$\dfrac{1}{2\pi}\int_{2\pi}\|X(\Omega)\|^2 d\Omega$

例 3-8 DTFT 时移性质说明

计算图 3.11 所示序列的 DTFT。此序列的数学描述为

$$x[n] = \delta[n-3] + 0.5\delta[n-4] + 0.25\delta[n-5] + \cdots = \sum_{k=3}^{\infty}(0.5)^{k-3}\delta[n-k]$$

$$= (0.5)^{n-3}u[n-3]$$

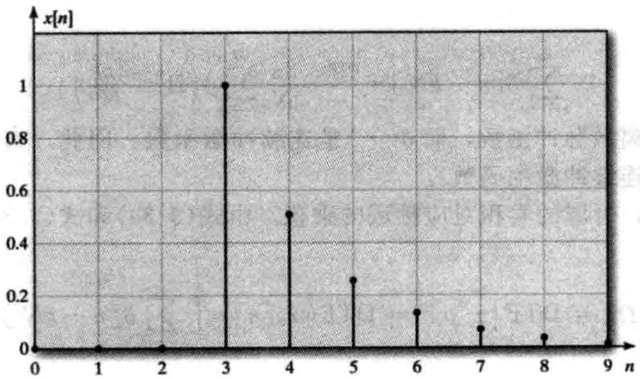


图 3.11 例 3-9 中的离散时间序列

由例 3-7 的结果可知，

$$\text{DTF}\{a^n u[n]\} = \frac{1}{1-ae^{-j\Omega}}$$

然后，由此变换结果和时移性质可知，

$$\text{DTF}\{(0.5)^{n-3}u[n-3]\} = \frac{e^{-j3\Omega}}{1-0.5e^{-j\Omega}}$$ ◀

例 3-9 系统脉冲响应的推导

离散系统的输入-输出关系可以由差分方程描述

$$y[n]=0.8y[n-1]+0.4x[n-1]$$

当输入为单位脉冲 $x[n]=\delta[n]$时，对应的输出定义为系统的脉冲响应，$y[n]=h[n]$。使用该标记重写差分方程为

$$h[n]-0.8h[n-1]=0.4\delta[n-1]$$

对差分方程进行变换，并利用的线性和时移性质得

$$H(\Omega)-0.8\mathrm{e}^{-\mathrm{j}\Omega}H(\Omega)=0.4\mathrm{e}^{-\mathrm{j}\Omega}$$

或

$$H(\Omega)=\frac{1}{\mathrm{e}^{\mathrm{j}\Omega}-0.7}=\mathrm{e}^{-\mathrm{j}\Omega}\left(\frac{\mathrm{e}^{\mathrm{j}\Omega}}{\mathrm{e}^{\mathrm{j}\Omega}-0.7}\right)$$

由表 3.5 可以看出，上式圆括号内因子的反变换是$(0.7)^n u[n]$。由表 3.6 可知，因子 $\mathrm{e}^{-\mathrm{j}\Omega}$ 对应信号的时移。因此系统脉冲响应为

$$h[n]=(0.7)^{n-1}u[n-1]$$ ◀

应用于本例中的情形并不多见，而将会更多采用 3.7 节介绍的 z 变换。后面将会发现，其与 z 变换紧密相关。

3.3.2 周期序列的 DTFT

对于周期序列的 DTFT，考虑周期为 N 的周期序列 $x[n]$，即 $x[n]=x[n+N]$。定义 $x[n]$从 $n=0$ 开始的一个周期内信号为 $x_0[n]$，即

$$x_0[n]=\begin{cases}x[n], & 0\leqslant n\leqslant N-1\\ 0, & \text{其他}\end{cases}\tag{3.33}$$

我们将发现，周期序列 $x[n]$的 DTFT 是 $X_0(\Omega)$的函数，$X_0(\Omega)$为 $x_0[n]$的 DTFT。

首先将周期序列 $x[n]$表示为

$$x[n]=x_0[n]*\sum_{k=-\infty}^{\infty}\delta[n-kN]\tag{3.34}$$

由于存在 DTFT 对

$$\sum_{k=-\infty}^{\infty}\delta[n-kN]\xleftrightarrow{\text{DTF}}\frac{2\pi}{N}\sum_{k=-\infty}^{\infty}\delta\left(\Omega-\frac{2\pi k}{N}\right)\tag{3.35}$$

注意，函数 $\delta[\cdot]$是离散脉冲函数，而 $\delta(\cdot)$是连续冲激函数。因此，离散周期脉冲序列的 DTFT 是频域中的连续冲激串函数。

由表 3.6 可知，时域的卷积对应频域的乘积。由式(3.34)和式(3.35)，$x[n]$的 DTFT 可以写为

$$X(\Omega)=\text{DTF}\{x[n]\}=\text{DTF}\left\{x_0[n]*\left[\sum_{k=-\infty}^{\infty}\delta[n-kN]\right]\right\}$$

即

$$X(\Omega)=\frac{2\pi}{N}\sum_{k=-\infty}^{\infty}X_0(\Omega)\delta\left(\Omega-\frac{2\pi k}{N}\right)\tag{3.36}$$

利用连续冲激函数的筛选性质，将式(3.36)写为

$$X(\Omega)=\frac{2\pi}{N}\sum_{k=-\infty}^{\infty}X_0\left(\frac{2\pi k}{N}\right)\delta\left(\Omega-\frac{2\pi k}{N}\right)\tag{3.37}$$

周期信号的 DTFT $X(\Omega)$的周期均为 2π。$x[n]$在同一个周期 $0\leqslant n\leqslant N-1$ 内的 N 个值转换成频域内 $X_0(2\pi k/N)$在 $0\leqslant k\leqslant N-1$ 中的 N 个值。

周期序列的DTFT是一个由离散频率分量组成的序列，该结果为离散傅里叶变换奠定了基础。

3.4 离散傅里叶变换

前面章节给出了傅里叶变换的计算公式。然而对于实际的傅里叶变换和离散时间傅里叶变换，经公式计算的结果都是频率的连续函数。本节将讨论傅里叶变换的近似计算方法，其利用连续时间信号有限的离散样本集来计算，得到有限的离散频率集。这种傅里叶变换的近似分析非常适用于数字计算机的计算，称为**离散傅里叶变换**(DFT)。

为了利用计算机计算连续时间信号的傅里叶变换，需将 $x(t)$进行抽样，抽样值以离散时间信号 $x[n]$的形式表示。由3.3节可得DTFT：

$$X(\Omega) = \mathrm{DTF}\{x[n]\} = \sum_{n=-\infty}^{\infty} x[n]\mathrm{e}^{-\mathrm{j}n\Omega} \qquad [(3.29)]$$

可依据离散时间样本计算DTFT，但 $X(\Omega)$是频率变量 Ω 的连续函数，不能由数字计算机直接表示，可通过计算连续频率函数的离散频率样本来近似分析 $X(\Omega)$。

考虑实际情况，生成离散频率样本必须限制离散时间样本为一个有限的集合。为了实现此目的，我们利用符号 N 表示所选取的离散时间信号的样本数。N 的取值要足够大，以使得样本集能够足以表示所有的 $x[n]$。通过将无限集的 $x[n]$与矩形窗函数相乘，可选择到所需的有限样本集[4]：

$$w_{\mathrm{R}}[n] = \begin{cases} 1, & n = 0,1,2,\cdots,N-1 \\ 0, & \text{其他} \end{cases}$$

用于计算频谱的有限样本集为

$$x_N[n] = x[n]w_{\mathrm{R}}[n] = \begin{cases} x[n], & n = 0,1,2,\cdots,N-1 \\ 0, & \text{其他} \end{cases} \qquad (3.38)$$

由式(3.29)可得信号 $x_N[n]$的频谱

$$X_N(\Omega) = \mathrm{DTF}\{x_N[n]\} = \sum_{n=-\infty}^{\infty} x_N[n]\mathrm{e}^{-\mathrm{j}n\Omega} = \sum_{n=0}^{N-1} x_N[n]\mathrm{e}^{-\mathrm{j}n\Omega} \qquad (3.39)$$

为了便于进一步讨论，认定 $x_N[n]$中含有 N 个样本的样本集为完整的信号。同时不再使用 $x_N[n]$中的下标，而是简化地用 $x[n]$来表示此有限离散时间序列。这也符合选择样本的有限集以足够表征整个信号的假设。

下面我们选择 $X_N(\Omega)$中的 N 个样本来表征频谱，可以选择多于 N 个样本，但是必须至少选择 N 个样本以避免在反变换时产生错误(详解参见参考文献3)。实际上，与表示 $x[n]$类似，通常选取相同数量的 N 个样本来计算频谱。由于DTFT的 Ω 周期为 2π，因此 N 个样本从同一个周期 $X_N(\Omega)$中获取。理想抽样后的频谱为

$$X_{\mathrm{S}}(\Omega) = X_N(\Omega)\sum_{k=0}^{N-1}\delta\left(\Omega - \frac{2\pi k}{N}\right) = \sum_{k=0}^{N-1} X_N\left(\frac{2\pi k}{N}\right)\delta\left(\Omega - \frac{2\pi k}{N}\right) \qquad (3.40)$$

$X_{\mathrm{S}}(\Omega)$中的各个抽样频率点的频谱值表现为频域冲激函数的权重。离散频率值 $X_N(2\pi k/N)$的集合是选择用来表示离散时间序列 $x[n]$频谱的信息。按照原先描述离散时间序列的习惯，将此离散频率序列表示为

$$X[k] = X_N(2\pi k/N), k = 0,1,2,\cdots,N-1$$

对比 $X_{\mathrm{S}}(\Omega)$与式(3.37)，频谱 $X_{\mathrm{S}}(\Omega)$与原先推导的周期信号DTFT具有相同的形式。(由于 $X[k]=0$，$k\neq 0, 1, 2, \cdots, N-1$，因此可以将 $X_{\mathrm{S}}(\Omega)$的公式中累加的上下限改为 $-\infty<k<\infty$。)实际上，如果利用式(3.30)计算 $X_{\mathrm{S}}(\Omega)$的反变换，将产生一个周期的离散时间序列 $x_{\mathrm{p}}[n]$，其在每个周期都有 N 个样本。然而，既然我们知道抽样后的频谱是由一个有限的、非周期的离散时间序列产生的，就可以将式(3.30)的反变换限制到 $x_p[n]$一个

周期内的 N 个样本。反变换所对应的离散时间信号为

$$x[n]=\frac{1}{N}\sum_{k=0}^{N-1}X[k]\mathrm{e}^{\mathrm{j}2\pi kn/N},\quad n=0,1,2,\cdots,N-1 \tag{3.41}$$

由离散时间信号 N 个样本集计算得到 N 个离散频率样本集的变换对分别称为离散傅里叶变换(DFT)和离散傅里叶反变换(IDFT)：

$$X[k]=\mathrm{DF}[x[n]]=\sum_{n=0}^{N-1}x[n]\mathrm{e}^{-\mathrm{j}2\pi kn/N},\quad k=0,1,2,\cdots,N-1 \tag{3.42}$$

$$x[n]=\mathrm{DF}^{-1}[X[k]]=\frac{1}{N}\sum_{k=0}^{N-1}X[k]\mathrm{e}^{\mathrm{j}2\pi kn/N},\quad n=0,1,2,\cdots,N-1 \tag{3.43}$$

符号 DF 代表离散傅里叶变换。同样地，利用 $x[n]\stackrel{\mathrm{DF}}{\longleftrightarrow}X[k]$ 表示离散傅里叶变换对。

例 3-10　计算离散傅里叶变换(DFT)

讨论三个数据点的离散时间信号，$N=3$，根据式(3.42)给出信号的离散傅里叶变换

$$\begin{aligned}X[0]&=x[0]+x[1]+x[2]\\X[1]&=x[0]+x[1]\mathrm{e}^{-\mathrm{j}2\pi/3}+x[2]\mathrm{e}^{-\mathrm{j}4\pi/3}\\X[2]&=x[0]+x[1]\mathrm{e}^{-\mathrm{j}4\pi/3}+x[2]\mathrm{e}^{-\mathrm{j}8\pi/3}\end{aligned} \tag{3.44}$$

第三个等式中的最后一项，$\mathrm{e}^{\frac{-\mathrm{j}8\pi}{3}}=\mathrm{e}^{\frac{-\mathrm{j}6\pi}{3}}\mathrm{e}^{\frac{-\mathrm{j}2\pi}{3}}=\mathrm{e}^{\frac{-\mathrm{j}2\pi}{3}}$，由式(3.43)得到离散傅里叶反变换

$$\begin{aligned}x[0]&=\frac{1}{3}(X[0]+X[1]+X[2])\\x[1]&=\frac{1}{3}(X[0]+X[1]\mathrm{e}^{\mathrm{j}2\pi/3}+X[2]\mathrm{e}^{\mathrm{j}4\pi/3})\\x[2]&=\frac{1}{3}(X[0]+X[1]\mathrm{e}^{\mathrm{j}4\pi/3}+X[2]\mathrm{e}^{\mathrm{j}8\pi/3})\end{aligned} \tag{3.45}$$

注意上面等式的对称性，这些等式通常由计算机处理，用来计算正向 DFT 的程序可以容易地修改为计算 IDFT，见(3.41)，并允许 N 为任意整数值。◀

3.4.1　DFT 的简写记号

在 DFT 中通常使用一个简写记号，定义符号 $W_N=\mathrm{e}^{-\mathrm{j}2\pi/N}$，式(3.42)和式(3.43)可表述为

$$X[k]=\mathrm{DF}[x[n]]=\sum_{n=0}^{N-1}x[n]W_N^{kn},\quad k=0,1,2,\cdots,N-1 \tag{3.46}$$

$$x[n]=\mathrm{DF}^{-1}[X[k]]=\frac{1}{N}\sum_{k=0}^{N-1}X[k]W_N^{-kn},\quad n=0,1,2,\cdots,N-1 \tag{3.47}$$

DFT 通常采用这种形式表示。

3.4.2　DTFT 的频率分辨率

由于离散信号的 DTFT 以 2π 为周期，对其一个周期内选取 N 个频率样本即为对应的 DFT，因此，利用 DFT 计算的离散频谱的分辨率(即样本之间的间隔)为

$$\Delta\Omega=\frac{2\pi}{N} \tag{3.48}$$

由式(3.48)可以看出，计算过程中 $x[n]$ 样本数 N 决定了频谱的分辨率，反之亦然，频谱要求的分辨率决定了所需的 $x[n]$ 样本数。如果时域样本数为 N_1，若要求更大的频域样本数 N_2 以提供足够的频率分辨率，可以在时间序列后附加 N_2-N_1 个零，该过程称为补零。在 3.5 节中讨论利用 DFT 计算线性卷积时，将说明补零的另一个应用。

例 3-11 **矩形脉冲的频谱**

利用 DFT 计算图 3.12a 所示的矩形脉冲的频谱。抽样周期为 $T=0.1(\mathrm{s})$ 以获得图 3.12b 所示的离散时间序列。MATLAB 程序如下：

```
% Create a time vector.
T = [-0.5:0.01:2];
% Create and plot a rectangular pulse centered at t = 0.5s.
x2 = rectpuls(T-0.5);
plot(T,x2)
% Create a discrete time sequence nT and a discrete-time
% rectangular pulse.
T2 = [0:0.1:1.9];
xs = rectpuls(T2-0.5);
xs = [0 0 0 0 0,xs]'
T2p = [-.5 -.4 -.3 -.2 -.1,T2];
% plot the discrete-time rectangular pulse sequence.
stem(T2p,xs)
stem(T2p,xs)
```

此程序绘制出与图 3.12a 和 b 相似的图，并生成离散时间序列 $x_s[n]$。

首先，此序列的 DFT 计算样本数为 $N=20$，$X[k]$的幅度使用下列 MATLAB 程序计算，并画在图 3.12c 中：

```
% Compute the DFT of xs[n] using (3.39).
N = 20;
W = exp(-j*2*pi/N)
X0 = 0
for k = 1:20;
  X0 = 0
  for n = 1:20;
    X(k) = xs(n)*W^((k-1)*(n-1))+X0;
    X0 = X(k);
  end
end
% Compute |X[k]|;
MX = abs(X);
% Create a vector of N plotting increments.
np = [0:length(X)-1];
% Plot |X[k]| vs. n.
stem(np,MX)
```

由式(3.48)可得，当 $N=20$ 时，频率分辨率为 $\Delta\Omega=2\pi/20=0.1\pi(\mathrm{rad})$ 或 $\Delta\omega=\Delta\Omega/T=\pi(\mathrm{rad/s})$。图 3.12c中的 $|X[k]|$ 的值在频率上以 $\pi(\mathrm{rad/s})$ 为间隔。

为了获得更好的频率分辨率，选择 $N=100$。在离散时间序列 $x_s[n]$后面补 80 个零。对补零后的序列进行 DFT 计算，图 3.12d 画出了 $|X[k]|$。在图 3.12d 中，频率分量之间的间隔是 $\Delta\Omega=2\pi/100=0.02\pi(\mathrm{rad})$或 $\Delta\omega=\Delta\Omega/T=0.2\pi(\mathrm{rad/s})$。◀

例 3-11 表明了合理选择计算 DFT 的样本数的重要性，以便获得足够的频谱分辨率。

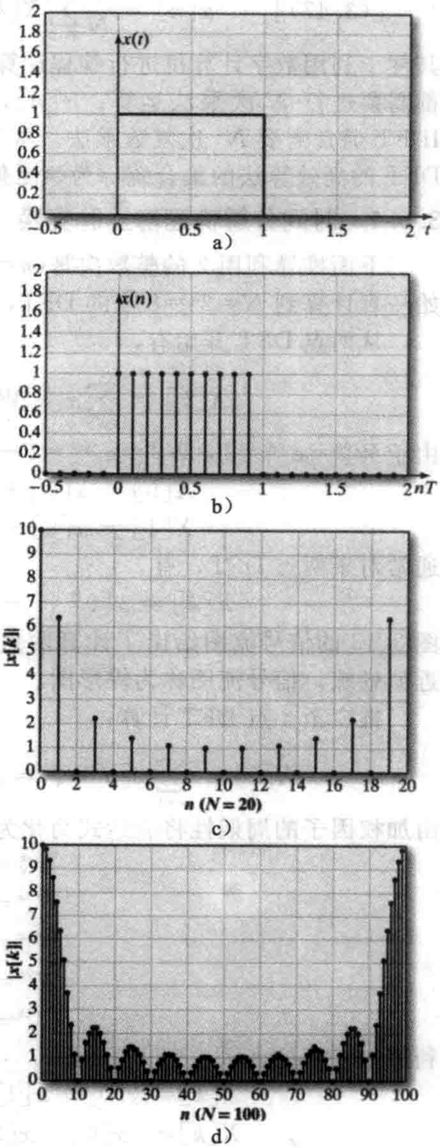


图 3.12 例 3-11 中的 DFT 图

3.4.3 章节小结

DFT 在数字信号处理中有着广泛的应用，本节讨论了 DFT 的数学表示和实际意义，以及计算 DFT 和 IDFT 的方法，并介绍了其典型应用。

在接下来的章节中，我们将学习计算离散时间序列 DFT 更加有效的方法，称为快速傅里叶变换。

3.5 快速傅里叶变换

3.4 节给出了 DFT 变换对公式，

[(3.46)] $$X[k]=\sum_{n=0}^{N-1}x[n]W_N^{kn},\quad k=0,1,2,\cdots,N-1$$

以及

[(3.47)] $$x[n]=\frac{1}{N}\sum_{k=0}^{N-1}X[k]W_N^{-kn},W_N=\mathrm{e}^{-\mathrm{j}2\pi/N},\quad n=0,1,2,\cdots,N-1$$

其便于利用数字计算机进行编程计算。观察式(3.46)可以发现对于每个 k 值，计算 $X[k]$ 都需要进行 N 次乘法运算。$x[n]$，尤其是 $X[k]$，都可能是复数，计算 N 点 DFT 或 IDFT 通常需要 N^2 次复数乘法。本节将推导更有效地计算 DFT 的算法。通常将用于计算 DFT 的高效算法的集合统称为快速傅里叶变换(FFT)。

3.5.1 时间分解快速傅里叶变换

下面推导利用 2 的整数次幂($n=2^m$)个样本来计算 DFT 的快速算法。首先从$N=2$开始一直计算到 $N=2^3=8$ 点的 DFT，该方法称为基 2 时间分解 FFT。

从两点 DFT 开始有，

$$X[k]=\sum_{n=0}^{1}x[n]W_2^{nk}=x[0]W_2^{0k}+x[1]W_2^{1k},\quad k=0,1$$

由于 $W_2^{0k}=\mathrm{e}^{-\mathrm{j}0}=1$，$W_2^{1k}=\mathrm{e}^{-\mathrm{j}\pi k}=(-1)^k$，写为

$$X[0]=x[0]+x[1]$$
$$X[1]=x[0]-x[1]$$

通常对于两点 DFT，有

$$X[k]=x[0]+(-1)^kx[1]$$

图 3.13 的信号流图给出了计算两点 DFT 的过程，由于形状近似蝴蝶，信号流图称为蝶形图。

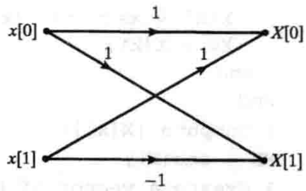


图 3.13 两点 DFT 蝶形图

推广至 4 点 DFT 计算：

$$X[k]=\sum_{n=0}^{3}x[n]W_4^{nk}=x[0]W_4^{0k}+x[1]W_4^{1k}+x[2]W_4^{2k}+x[3]W_4^{3k}$$

由加权因子的周期性将表达式简化为

$$W_N^{nk}=\mathrm{e}^{-\mathrm{j}(2\pi/N)nk}$$
$$W_4^{0k}=1$$
$$W_4^{1k}=\mathrm{e}^{-\mathrm{j}(\pi/2)k}=(-\mathrm{j})^k$$
$$W_4^{2k}=\mathrm{e}^{-\mathrm{j}\pi k}=(-1)^k$$
$$W_4^{3k}=W_4^{2k}W_4^{1k}=(-1)^kW_4^{1k}$$

利用以上简化表示，有

$$X[k]=x[0]+x[1]W_4^{1k}+x[2](-1)^k+x[3](-1)^kW_4^{1k}$$
$$X[k]=\{x[0]+x[2](-1)^k\}+\{x[1]+x[3](-1)^k\}W_4^{1k}$$

为了简化下面的步骤，定义两个新变量：

$$x_e[n] = x[2n],\quad n = 0,1$$
$$x_o[n] = x[2n+1],\quad n = 0,1$$

因此

$$X[k] = \{x_e[0] + x_e[1](-1)^k\} + \{x_o[0] + x_o[1](-1)^k\}W_4^{1k}$$

等式中括号内的因子可看做是两点 DFT：

$$X_e[m] = x_e[0] + x_e[1](-1)^m, m = 0,1$$
$$X_o[m] = x_o[0] + x_o[1](-1)^m, m = 0,1$$

由于 $X_e[k]$和 $X_o[k]$是周期的，

$$X_e[2] = x_e[0] + x_e[1](-1)^2 = X_e[0]$$
$$X_e[3] = x_e[0] + x_e[1](-1)^3 = X_e[1]$$

因此，4 点 DFT 为

$$\begin{aligned} X[0] &= x_e[0] + x_o[0]W_4^{1(0)} = x_e[0] + x_o[0] \\ X[1] &= x_e[1] + x_o[1]W_4^{1(1)} = x_e[1] + x_o[1]W_4^1 \\ X[2] &= x_e[0] - x_o[0]W_4^{1(2)} = x_e[0] - x_o[0] \\ X[3] &= x_e[1] - x_o[1]W_4^{1(3)} = x_e[1] - x_o[1]W_4^1 \end{aligned} \tag{3.49}$$

其中的变化项是随 k 而变化的 $W_4^{1(k)}$。

由此可以看出，4 点 DFT 可由两点 DFT 计算，如图 3.14 中的信号流图所示。换言之，

$$\{x[n]4\text{ 点 DFT}\} = \{x_e[n]\text{ 两点 DFT}\} + W_4^{1(k)}\{x_o[n]\text{ 两点 DFT}\}$$

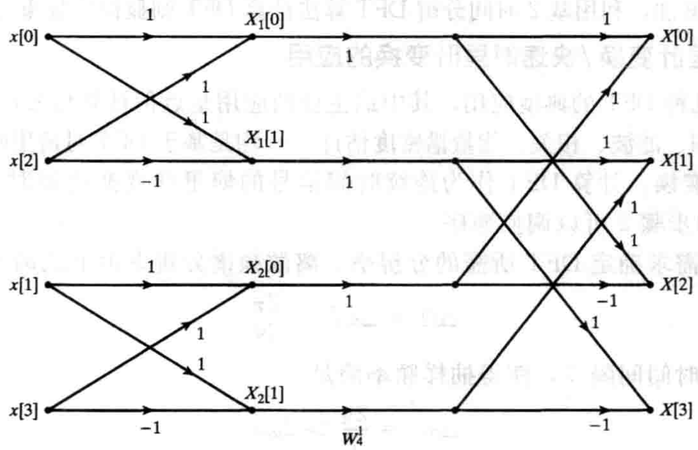


图 3.14　4 点 DFT 信号流图

式(3.49)称为 4 点 DFT 的**重组方程**。

下面推导 8 点 DFT 的计算，

$$X[k] = \sum_{n=0}^{7} x[n]W_8^{nk},\quad k = 0,1,\cdots,7$$

8 点 DFT 的加权因子为

$$\begin{aligned} W_8^{0k} &= 1 \\ W_8^{1k} &= e^{-j(\pi/4)k} \\ W_8^{2k} &= e^{-j(\pi/4)2k} = e^{-j(\pi/2)k} = W_4^{1k} \\ W_8^{3k} &= e^{-j(\pi/4)3k} = [e^{-j(\pi/4)2k}]e^{-j(\pi/4)k} = W_8^{1k}W_4^{1k} \\ W_8^{4k} &= e^{-j(\pi/4)4k} = e^{-j\pi k} = W_4^{2k} \end{aligned}$$

$$W_8^{5k} = e^{-j(\pi/4)5k} = e^{-j(\pi/4)4k}e^{-j(\pi/4)k} = W_8^{1k}W_4^{2k}$$
$$W_8^{6k} = e^{-j(\pi/4)6k} = W_4^{3k}$$
$$W_8^{7k} = e^{-j(\pi/4)7k} = e^{-j(\pi/4)k}e^{-j(\pi/4)6k} = W_8^{1k}W_4^{3k}$$

利用上述加权因子将 8 点 DFT 写为

$$X[k] = x[0] + x[1]W_8^{1k} + x[2]W_4^{1k} + x[3]W_8^{1k}W_4^{1k} + x[4]W_4^{2k} + x[5]W_8^{1k}W_4^{2k} + x[6]W_4^{3k} + x[7]W_8^{1k}W_4^{3k} \tag{3.50}$$

如同推导 4 点 DFT，定义

$$x_e[n] = x[2n], \quad n = 0,1,2,3$$

及

$$x_o[n] = x[2n+1], \quad n = 0,1,2,3$$

将式(3.50)重写为

$$X[k] = \{x_e[0] + x_e[1]W_4^{1k} + x_e[2]W_4^{2k} + x_e[3]W_4^{3k}\} + W_8^{1k}\{x_o[0] + x_o[1]W_4^{1k} + x_o[2]W_4^{2k} + x_o[3]W_4^{3k}\}$$

花括号内的因子分别是 $x_e[n]$和 $x_o[n]$的 4 点 DFT，即 8 点 DFT 是由两个 4 点 DFT 计算得到。

通常，N 点基 2 时间分解 DFT 可由两个$N/2$点的 DFT 计算。

如前所述，推导 DFT 算法的原因是为了提高 DFT 的计算效率。直接计算 N 点 DFT 的乘法复杂度通常为 N^2，而利用基 2 时间分解 DFT 算法，乘法复杂度可以降低至$(N/2)\log_2(N)$。随着 DFT 点数的增加，利用基 2 时间分解 DFT 算法计算 DFT 则显得更加重要。

3.5.2　离散傅里叶变换/快速傅里叶变换的应用

本节将介绍几种 DFT 的典型应用，其中最主要的应用是近似计算信号的傅里叶变换，其他的应用——卷积、滤波、相关、能量谱密度估计——均是基于 DFT 对傅里叶变换的近似。

计算傅里叶变换。计算 DFT 作为连续时间信号的傅里叶变换近似时，需要以下几个步骤：(步骤 1 和步骤 2 可以调换顺序。)

1. 针对特定需求确定 DFT 所需的分辨率。离散频谱分辨率由下式确定

$$\Delta\Omega = \Delta\omega T = \frac{2\pi}{N} \tag{3.51}$$

2. 确定抽样时间间隔 T，使得抽样频率满足

$$\Delta\omega_s = \frac{2\pi}{T} > 2\omega_M$$

其中，ω_M 是连续时间信号的最高频率分量。(理由将在第 4 章中详细讨论。)

3. 在 NT 秒的时间段内采集连续时间信号的 N 个样本。如果步骤 4 采用基 2 DFT 方法计算 DFT，则 N 必须是 2 的整数次幂。在某些场合下，若 N 只能取固定值，可通过补零使样本数 N 达到 2 的整数次幂。

4. 计算 DFT，可直接利用式(3.46)进行计算，但更常用且更有效率的计算方法是采用 DFT 快速算法。

正确执行以上步骤所得到 DFT 的结果应是连续时间信号傅里叶变换的良好近似。下面通过讨论几个例子说明处理过程，并指出需要特别注意的某些细节。

例 3-12　利用近似计算傅里叶变换

计算图 3.15a 所示的矩形脉冲傅里叶变换。信号可以描述为

$$x(t) = u(t) - u(t-1)$$

由表3.3，令 $T_R=1$，利用表3.4中的时移性质，令 $t_0=0.5$s，可得此信号的傅里叶变换为

$$X(\omega)=\operatorname{sinc}(\omega/2)\mathrm{e}^{-\mathrm{j}\omega/2}$$

图3.15b给出了 $X(\omega)$的幅度曲线，其第一个零值出现在 $\omega=2\pi$ rad/s处。尽管该频谱没有一个绝对的带宽，但可以看到旁瓣幅度随着频率增加而逐渐衰减。选择抽样间隔 $T=0.1$s，抽样频率 $\omega_s=20\pi$ rad/s，即第一个零值点频率的10倍。

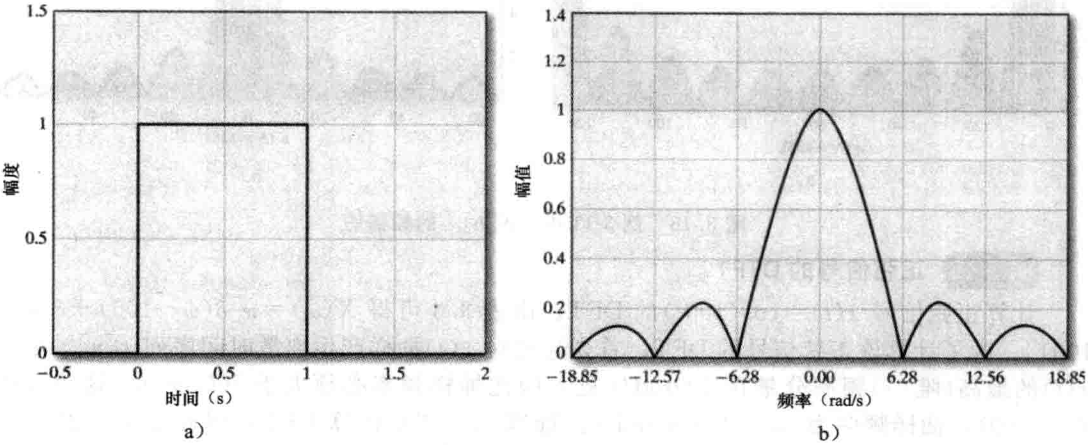


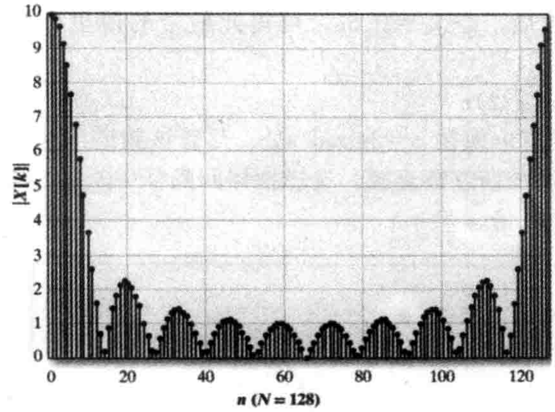
图3.15 例3-12的图

采用 $N=128$ 的基2 FFT算法近似计算矩形脉冲的频谱。由式(3.51)可知，$N=128$ 产生的频率分辨率为 $\Delta\Omega=2\pi/128=0.0156\pi$ rad，或 $\Delta\omega=\Delta\Omega/T=0.156\pi$ rad/s。下面的MATLAB程序给出频谱计算的结果。

```
% Create a sequence of sample time values. with sample period T = 0.1 s.
T2=[0:0.1:1.9];
% Create the discrete-time rectangular pulse u[n]-u[n-1] of sample %values.
xs=rectpuls(T2-0.5);
% Zero-pad the sample sequence so that N = 128.
m=zeros(1,108);
xs=[xs,m];
% Compute the DFT using the radix-2 FFT algorithm.
Xs=fft(xs,128);
% Plot the magnitude frequency spectrum of DFT values.
figure(1), stem(abs(Xs))
% Use the FFTSHIFT function of MATLAB to center the frequency spectrum
%around n = 0.
Xks=fftshift(Xs);
np=[-64:63];
figure(5),stem(np,abs(Xks));
```

图3.16a显示了由MATLAB函数fft生成的DFT幅度图。图3.16b显示了由MATLAB函数fftshift进行数据平移后的DFT幅度图，其使得零频分量处于中间位置。

由于DFT由离散时间样本计算所得，因此需要将 $|X[k]|$ 乘以周期 T 以抵消信号抽样后傅里叶变换的固有因子 $1/T$。(抽样后信号傅里叶变换的 $1/T$ 因子将在第4章中讨论。)如果图3.16b的幅度乘以 $T=0.1$，离散的自变量乘以 $\Delta\omega=\Delta\Omega/T=0.156\pi$，则由DFT算法计算得到的频谱近似为图3.15中连续时间信号的频谱。 ◀



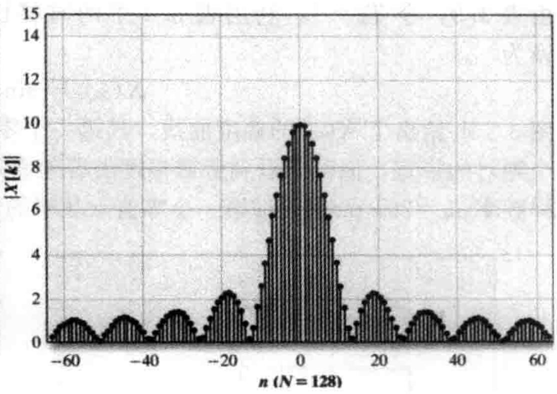


图 3.16 例 3-11 中 $|X[k]|$ 的频谱图

例 3-13 正弦信号的 DTFT

计算正弦信号 $x(t)=\cos(100t)$ 的 DFT。由表 3.3 可得 $X(\omega)=\pi[\delta(\omega-100)+\delta(\omega+100)]$。为了计算该正弦信号的 DFT，首先通过对 $x(t)$ 抽样产生离散时间序列 $x[n]$。由于 $x(t)$ 的最高(唯一)频率分量在 100rad/s 处，因此抽样频率必须大于 200rad/s。这里选择 $T=0.001$，抽样频率为 $\omega_s=2000\pi$ rad/s。选取 $N=256$ 计算 DFT，因此 $\Delta\omega=\Delta\Omega/T=2\pi/(NT)=24.544$ rad/s，用于产生抽样值序列和计算 DFT 的 MATLAB 程序如下：

```
% Create a discrete-time vector of 256 sample times with
%T = 0.001s.
t=[0:0.001:0.255];
% Create the discrete-time sequence of sample values
xn=cos(100*t);
% Compute the DFT of the sample sequence using the FFT algorithm
Xk=fft(xn);
% FFTSHIFT places the zero frequency component in the center.
Xks=fftshift(Xk);
% Plot the magnitude of the DFT components.
np=[-128:127];
stem(np,abs(Xks))
```

图 3.17 画出了计算出的 DFT 频率分量的幅度。可以看到，DFT 主要的频率分量在 $\omega=4\times\Delta\omega\approx100$ rad/s处存在，同时在其他 $\Delta\omega$ 整数倍处也存在频率分量。

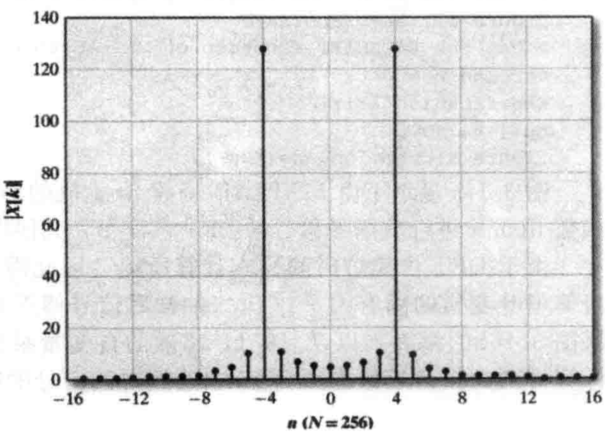


图 3.17 例 3-12 的 DFT 图 ◀

由例 3-13 中可以看出，在利用 DFT 近似计算连续时间信号频谱的过程中，隐含了“加窗”处理和信号的周期延拓，加窗是一个误差源。将序列 $x[n]$ 截短的操作相当于把序列 $x[n]$ 乘以一个宽度为 N 的矩形序列。抽样点序列 $x[n]$ 与窗函数时域相乘所导致的信号频谱扩散的现象称为频谱泄漏失真(Spectrum-leakage distortion)，这由截断信号而出现的频谱延拓造成。

频谱泄漏失真现象可以由图 3.18a 所示截短余弦信号来说明，信号由余弦波和图 3.18b 所示矩形脉冲的乘积

得到，根据前面傅里叶变换的分析可知

$$\cos(\omega_1 t) \overset{F}{\longleftrightarrow} \pi\{\delta[\omega-\omega_1]+\delta[\omega+\omega_1]\}$$

及如图 3.18c 所示的

$$\left[u\left(t+\frac{T}{2}\right)-u\left(t-\frac{T}{2}\right)\right] \overset{F}{\longleftrightarrow} T\mathrm{sinc}\left(\frac{T\omega}{2}\right)$$

由傅里叶变换时域乘积(或频率卷积)性质

$$x_1(t)x_2(t) \overset{F}{\longleftrightarrow} \frac{1}{2\pi}X_1(\omega) * X_2(\omega)$$

截断的余弦信号频谱如图 3.18d 所示。由此可以看到，截断余弦信号的频谱扩展到所有频率范围。正是由信号的截短造成的频谱扩展导致了例 3-12中的 DFT 频谱泄漏失真。

除了矩形窗之外，其他几种可选的窗口类型可用来降低频谱泄露失真。其中两种最著名的窗函数就是由 Richard W. Hamming 提出的汉明窗和以 Julius von hann 命名的汉宁窗。

汉明窗

$$w[n]=0.54-0.46\cos\left(\frac{2\pi n}{N-1}\right),\quad 0\leqslant n\leqslant N-1 \tag{3.52}$$

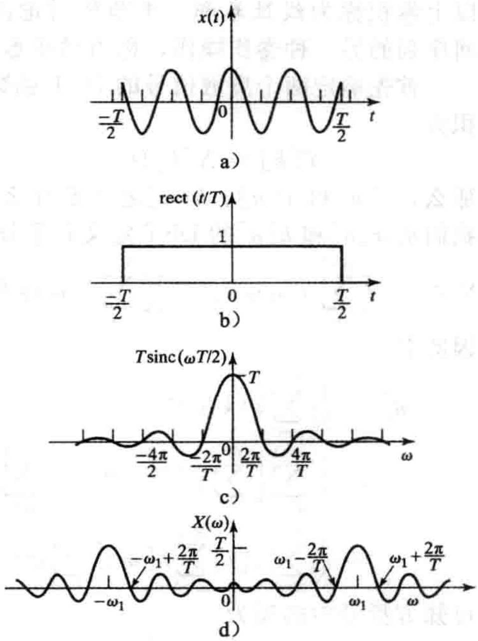


图 3.18 加窗造成的频谱扩展效应

汉宁窗

$$w[n]=0.50-0.50\cos\left(\frac{2\pi n}{N-1}\right),\quad 0\leqslant n\leqslant N-1 \tag{3.53}$$

这两个窗函数都在窗口的末端逐渐趋近于零，而不是像矩形窗那样突然切断。这种时域的渐变导致了其频域中更窄的带宽，因此表现在 DFT 中将产生更少的谱泄漏失真。

例 3-14 汉明窗的应用

本例是例 3-13 的扩展。使用汉明窗截断数据，而不是例 3-13 中隐含的矩形窗。由 MATLAB 程序来完成计算：

```
% Create the Hamming Window with N = %256 and %multiply it by
the %discrete-time sequence x[n].
xham=hamming(256)'.*xn
% Compute the DFT of the Hamming %windowed sequence.
Xk=fft(xham);
% Plot the DFT with frequency %components centered at zero %frequency.
Xks=fftshift(Xk);
np=[0:length(Xk)-1];
stem(np,abs(Xks))
```

由 DFT 得到的近似频谱如图 3.19 所示。可以看到，如同例 3-13，有效频率分量在 $n=\pm3$和 $n=\pm4$ 处，主要频率在 $n=\pm4$ 处，而相比较例 3-13，其频谱扩展显著减少。◀

由以上例子可以看出，DFT 可以给出傅里叶变换很好的近似，通过选择合适的窗函数可以减少频谱泄漏失真以降低傅里叶变换近似中的误差。计算过程中提高抽样率和增加样本数目同样也可以减少误差，但这将导致计算效率的降低。

利用 DFT/FFT 计算卷积。 两个离散时间信号的卷积由式(2.78)描述为

$$y[n]=\sum_{k=-\infty}^{\infty}x[k]h[n-k]=x[n]*h[n]$$

利用傅里叶变换的卷积性质可得

$$Y(\Omega) = X(\Omega)H(\Omega)$$

以上卷积称为线性卷积。本节将讨论离散时间序列的另一种卷积操作，称为循环卷积。

首先确定两个离散信号的 DFT 函数的乘积为

$$Y[k] = X[k]H[k]$$

那么，$y[n]$和 $x[n]$、$h[n]$之间是什么关系？我们从 $x[n]$和 $h[n]$的 DFT 定义着手分析，

$$Y[k] = \left[\sum_{n=0}^{N-1} x[n]\mathrm{e}^{-\mathrm{j}2\pi kn/N}\right]\left[\sum_{m=0}^{N-1} h[m]\mathrm{e}^{-\mathrm{j}2\pi km/N}\right]$$

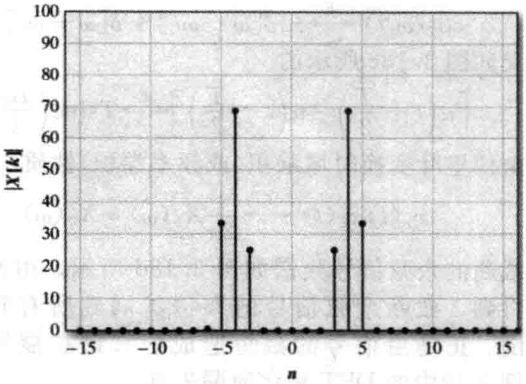


图 3.19　例 3-13 的 DFT 图

因此有

$$\begin{aligned} y[n] &= \frac{1}{N}\sum_{k=0}^{N-1} Y[k]\mathrm{e}^{\mathrm{j}2\pi kn/N} \\ &= \frac{1}{N}\sum_{k=0}^{N-1}\left[\sum_{l=0}^{N-1} x[l]\mathrm{e}^{-\mathrm{j}2\pi kl/N}\right]\left[\sum_{m=0}^{N-1} h[m]\mathrm{e}^{-\mathrm{j}2\pi km/N}\right]\mathrm{e}^{\mathrm{j}2\pi kn/N} \\ &= \frac{1}{N}\sum_{l=0}^{N-1} x[l]\sum_{m=0}^{N-1} h[m]\left[\sum_{k=0}^{N-1}\mathrm{e}^{\mathrm{j}2\pi k(n-l-m)/N}\right] \end{aligned}$$

可知方括号内的项为

$$\sum_{k=0}^{N-1}\mathrm{e}^{\mathrm{j}2\pi k(n-l-m)/N} = \begin{cases} N, & n-l-m=0 \\ 0, & \text{其他} \end{cases}$$

因此有，

$$y[n] = \frac{1}{N}\sum_{l=0}^{N-1} x[l]\sum_{m=0}^{N-1} h[m]N\delta[n-l-m]$$

由于脉冲函数 $\delta[n-l-m]$只在 $m=n-l$ 处不为零，重写等式为

$$y[n] = \sum_{l=0}^{N-1} x[l]h[n-l] \tag{3.54}$$

显然，其与线性卷积表达式相似。然而，上式累加仅在一个周期内，而不是在所有时间内。该等式代表的过程称为周期卷积或者循环卷积，本书通常采用后者来命名。

利用 DFT 计算线性卷积　长度分别为 N_1 和 N_2 两个序列的线性卷积，等于这两个序列补零至 N_1+N_2-1 个样本后的循环卷积。

例 3-15　补零的循环卷积

计算两个序列的线性卷积

$$x[n] = [1,2,3,4] \text{ 及 } h[n] = [0,1,2,3])$$
$$y[n] = x[n] * h[n] = [0,1,4,10,16,17,12]$$

利用两个序列的 DFT，计算其循环卷积的 MATLAB 程序和结果如下：

```
Xn=fft(xn);
Hn=fft(hn);
Yn=Xn.*Hn
yn=ifft(Yn)
```

结果为

$$x[n] \otimes h[n] = [16,18,16,10]$$

对每个序列首先补零使得长度为 $N_1+N_2-1=7$，再通过循环卷积计算线性卷积，

$$x_p[n] = [1,2,3,4,0,0,0] \text{ 及 } h_p[n] = [0,1,2,3,0,0,0]$$

$x_p[n]$和 $y_p[n]$循环卷积的 DFT 为

$$Y_p[k]=X_p[k]H_p[k]=[60,-19.9928+\mathrm{j}23.8775,-5.6024+\mathrm{j}0.3927,-5.8342-\mathrm{j}0.8644-4.4049+\mathrm{j}0.4586,-5.6024-\mathrm{j}0.3927,-19.9928+\mathrm{j}23.8775]$$

其 IDFT 给出了 $x_p[n]$和 $y_p[n]$的循环卷积，其与 $x[n]$和 $h[n]$的线性卷积的结果相同。

$$y_p[n]=\frac{1}{7}\sum_{k=0}^{7}Y_p[k]\mathrm{e}^{\mathrm{j}2\pi kn/7},n=0,1,2,\cdots,7$$

$$y_p[n]=y[n]=[0,1,4,10,16,17,12]$$

获得此线性卷积结果的 MATLAB 程序为

```
% zero pad x[n] and h[n] to have N1 + N2 - 1 elements.
xnp=[xn,0 0 0]; hnp=[hn,0 0 0];
% Compute the DFT's of the two zero-padded sequences and multiply them
together.
Xp=fft(xnp);
Hp=fft(ynp);
Y=Xp.*Hp;
% Compute the inverse DFT to obtain the convolution result.
y=ifft(Y)
```

对序列补零的过程使得循环卷积结果与原始序列的线性卷积结果相同，这在信号处理应用中非常重要，补零过程允许利用两个序列的 DFT 计算其对应的线性卷积。◀

计算效率 现在考虑采用 DFT 而非直接方法计算线性卷积的计算效率。图 3.20 为利用 DFT 方法计算卷积过程的框图，此系统中 DFT 和 IDFT 均采用基 2 DFT 计算，所涉及的两个 N 点序列都通过补零进行扩展，使得每个扩展后的序列长度为 N_1+N_2-1。为了简化代数计算，同时避免严重误差，我们假设两个序列的长度都为 N，这样每个序列补零后的长度近似为 $2N$。如果假设离散时间样本是复数，则计算每个补零后序列的 DFT 需要 $N\log_2(2N)$次复数乘法。快速离散傅里叶反变换(IFFT)的计算同样也需要 $N\log_2(2N)$次复数乘法。两个序列的 DFT 相乘 $X_1[k]\times X_2[k]$需要 $2N$ 次复数乘法。由此可见，利用 DFT 方法计算两个 N 点序列的线性卷积共计需 $3N\log_2(2N)+2N$ 次复数乘法运算。

两个序列的直接卷积需要 N^2 次复数乘法。如果假设选择 N 为 2 的幂次，则可以利用基 2 FFT。当 $N\geqslant 32$ 时，DFT 方法更有效。

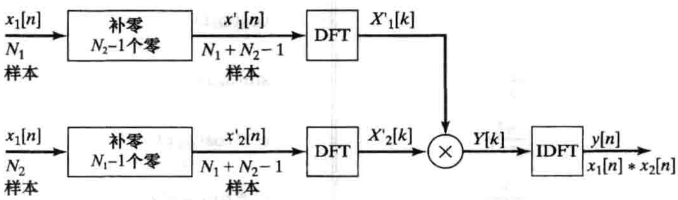


图 3.20 利用 DFT 计算线性卷积的框图

3.6 拉普拉斯变换

可以认为拉普拉斯变换是傅里叶变换的推广，本节研究拉普拉斯变换的原因在于其在数字滤波器设计中的重要作用。许多递归数字滤波器的设计均是基于连续时间(模拟)滤波器，通常由模拟滤波器设计开始，利用连续时间频域到离散时间频域之间的映射来构造一个递归数字滤波器。经典的模拟滤波器(巴特沃兹、切比雪夫和椭圆滤波器)通过映射都有对应的数字滤波器。

式(3.18)给出的傅里叶变换的表示式为

$$X(\omega) = \int_{-\infty}^{\infty} x(t) \cdot e^{-j\omega t} dt$$

如同 z 变换的处理过程，注意，上式中 $e^{-j\omega t}$ 是个比较特别的复数，其幅度总为 1。如果用一个可以在复平面上取任意值的复变量 $s=\sigma+j\omega$ 来替换该复数，可得

$$X(s) = \int_{-\infty}^{\infty} x(t) \cdot e^{-st} dt \tag{3.55}$$

这就是拉普拉斯变换的定义式，对应的拉普拉斯反变换为

$$x(t) = \frac{1}{2\pi j}\int_{\sigma-j\infty}^{\sigma+j\infty} X(s) \cdot e^{st} ds \tag{3.56}$$

式(3.55)和式(3.56)定义了双边拉普拉斯变换。

由于模拟滤波器是因果的，因此很容易定义单边拉普拉斯变换。该变换的定义式为

$$X(s)= \int_{0}^{\infty} x(t)e^{-st} dt \tag{3.57}$$

$$x(t)= \frac{1}{2\pi j}\int_{\sigma-j\infty}^{\sigma+j\infty} X(s)e^{st} ds \tag{3.58}$$

由于连续时间信号难以存储(因此构建伪非因果系统)，因此单边拉普拉斯变换比双边拉普拉斯变换有着更多的应用。本书中提到的拉普拉斯变换指的是单边拉普拉斯变换。如果用到双边拉普拉斯变换，本书将会明确指出。

拉普拉斯变换可视为信号乘以指数信号后的傅里叶变换。由于 $s=\sigma+j\omega$，因此可将 $x(t)$ 的双边拉普拉斯变换写为

$$X(s) = \int_{-\infty}^{\infty} [x(t)e^{-\sigma t}]e^{-j\omega t} dt$$

我们发现，其为信号 $(x(t)e^{-\sigma t})$ 的傅里叶变换。因此，拉普拉斯变换的优势在于其收敛域(积分存在对应的积分区域)比傅里叶变换的收敛域大得多。实际上，很多不存在傅里叶变换的信号存在拉普拉斯变换。

表 3.7 列出了部分连续时间函数的拉普拉斯变换。

表 3.7 拉普拉斯变换对

$x(t)$, $t\geqslant 0$	$X(s)$	$x(t)$, $t\geqslant 0$	$X(s)$
$\delta(t)$	1	$\cos(\omega_0 t)$	$\frac{s}{s^2+\omega_0^2}$
$u(t)$	$\frac{1}{s}$	$\sin(\omega_0 t)$	$\frac{\omega_0}{s^2+\omega_0^2}$
t^n	$\frac{n!}{s^{n+1}}$	$e^{-at}\cos(\omega_0 t)$	$\frac{s+a}{(s+a)^2+\omega_0^2}$
e^{-at}	$\frac{1}{s+a}$	$e^{-at}\sin(\omega_0 t)$	$\frac{\omega_0}{(s+a)^2+\omega_0^2}$
$t^n e^{-at}$	$\frac{n!}{(s+a)^{n+1}}$		

3.6.1 拉普拉斯变换的性质

拉普拉斯变换有很多重要的性质，其中的三条性质对于滤波器的分析和设计非常关键，它们是线性、卷积和微分。

拉普拉斯变换的线性性质可表示为

如果

$$x_1(t) \longleftrightarrow X_1(s), x_2(t) \longleftrightarrow X_2(s)$$

则

$$ax_1(t)+bx_2(t)\longleftrightarrow aX_1(s)+bX_2(s)$$

线性性质允许将简单信号的拉普拉斯变换组合成为复杂信号的变换，通过将函数拆分为多个较为简单的函数，如表3.7，查找简单函数以实现反变换。因此，很少借助拉普拉斯反变换的积分公式进行计算。

表3.8列出了拉普拉斯变换的主要性质。

表3.8 拉普拉斯变换性质

操作	时间函数	拉普拉斯变换
线性	$ax_1(t)+bx_2(t)$	$aX_1(s)+bX_2(s)$
时移	$x(t-t_0)u(t-t_0)$	$X(s)\mathrm{e}^{-t_0 s}$
时域展缩	$x(at)$	$\frac{1}{\lvert a\rvert}X\left(\frac{s}{a}\right)$
时域变换	$x(at-t_0)u(at-t_0)$	$\frac{1}{\lvert a\rvert}X\left(\frac{s}{a}\right)\mathrm{e}^{-t_0 s/a}$
复数偏移	$x(t)\mathrm{e}^{-at}$	$X(s+a)$
时域卷积	$x_1(t)*x_2(t)$	$X_1(s)X_2(s)$
s域卷积	$x_1(t)\cdot x_2(t)$	$X_1(s)*X_2(s)$
时域微分	$\frac{\mathrm{d}^n x(t)}{\mathrm{d}t^n}$	$s^nX(s)-s^{n-1}x(0^-)-\cdots-sx^{(n-2)}(0^-)-x^{(n-1)}(0^-)$
复频域微分	$t^n x(t)$	$(-1)^n\frac{\mathrm{d}^n X(s)}{\mathrm{d}s^n}$
时域积分	$\int_0^t x(\tau)\mathrm{d}\tau$	$\frac{1}{s}X(s)$
时域周期	$x_p(t)=\sum_{n=0}^{\infty}x(t-nT_0)$	$X_p(s)=\frac{1}{1-\mathrm{e}^{-T_0 s}}X(s)$
初值	$\lim_{t\to 0}x(t)$	$\lim_{s\to\infty}sX(s)$
终值	$\lim_{t\to\infty}x(t)$	$\lim_{s\to 0}sX(s)$

卷积性质说明了时域卷积对应频率乘积。当两个系统级联时，一个系统的输出是另一系统的输入，必须在时域通过卷积计算，才可以根据第一个系统的输入，求解出第二个系统的输出。采用系统的拉普拉斯变换时，利用卷积性质可将卷积运算转换为乘积运算。微分性质说明了时域微分对应拉普拉斯变换乘以 s。通常，对于 n 阶微分和单边拉普拉斯变换有

$$\frac{\mathrm{d}^n x(t)}{\mathrm{d}t^n}\longleftrightarrow s^nX(s)-s^{n-1}x(0^-)-\cdots-sx^{n-2}(0^-)-x^{n-1}(0^-)$$

对于 s 域中传输函数初始条件认为是零。相反地，时域积分对应复频域除以 s。

微分性质使得可将微分方程转换为代数方程，并且将线性非时变系统的输入输出关系表示为系统的传输函数。

3.6.2 传输函数

连续时间系统的传输函数定义为所有初始条件均为零时，输出信号的拉普拉斯变换与

输入信号的拉普拉斯变换之比。因此，根据描述模拟滤波器输入-输出关系的微分方程，利用拉普拉斯变换及其微分性质可得到该模拟滤波器的传输函数。

例 3-16 传输函数的推导

连续时间系统由微分方程描述

$$\frac{\mathrm{d}^2 y}{\mathrm{d}t^2}+754\frac{\mathrm{d}y}{\mathrm{d}t}+142000y(t)=142000x(t)$$

为推导该系统的传输函数，利用线性和时域微分性质，逐项写出微分方程的拉普拉斯变换

$$s^2Y(s)+754sY(s)+142000Y(s)=142000X(s)$$

提取出左侧每项中的 $Y(s)$ 可得

$$(s^2+754s+142000)Y(s)=142000X(s)$$

因此，该系统的传输函数为

$$H(s)=\frac{Y(s)}{X(s)}=\frac{142000}{s^2+754s+142000}$$ ◀

3.6.3 连续时间LTI系统的频率响应

只需通过用 $\mathrm{j}\omega$ 替换系统传输函数中的 s，即可得到系统的频率响应。

例 3-17 频率响应求解

求解模拟滤波器的频率响应，其传输函数为

$$H(s)=\frac{142\times10^3}{s^2+754s+142\times10^3}$$

解： 为得到该传输函数对应的频率响应，利用 $\mathrm{j}\omega$ 替换 s 得到

$$H(\mathrm{j}\omega)=\frac{142\times10^3}{142\times10^3-\omega^2+\mathrm{j}754\omega}$$

由上式写出其幅度谱和相位谱为

$$|H(\mathrm{j}\omega)|=\frac{142\times10^3}{\sqrt{(142\times10^3-\omega^2)^2+(754\omega)^2}},\quad \angle H(\mathrm{j}\omega)=0-\arctan\left(\frac{754\omega}{142\times10^3-\omega^2}\right)$$

幅度谱和相位谱如图 3.21 所示。

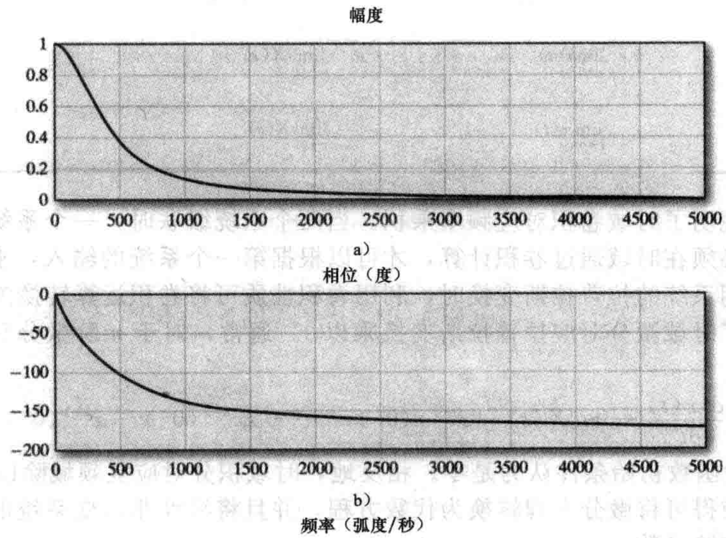


图 3.21 例 3-13 的幅度谱和相位谱

图 3.21 由如下 MATLAB 程序生成：

```
n=142e+3
d=[1 754 n]
W=[0:10:5000];
[FR,W]=freqs(n,d,W);
subplot(2,1,1), plot(W,abs(FR)),grid
subplot(2,1,2),plot(W,angle(FR)*180/pi),grid
```

◀

3.7 z 变换

在线性时不变系统的分析和设计中，各种变换起到了重要作用，极大简化了所需的数学计算。如变换使得微分方程求解转变为代数方程的求解，将卷积运算转变为乘积运算。对于数字滤波器而言，离散时间傅里叶变换(DTFT)和离散傅里叶变换(DFT)尤其重要。然而，某些信号由于变换不收敛而不存在变换。同时，DTFT 和 DFT 的标记也比较繁琐。本节将注意力转移到 z 变换，可视为 DTFT 的推广。z 变换和 DTFT 的联系与拉普拉斯变化和傅里叶变换的联系极其相似。

DTFT 定义为 $X(\Omega)=\sum\limits_{n=-\infty}^{\infty}x[n]\mathrm{e}^{-\mathrm{j}n\Omega}$，式中，$\mathrm{e}^{-\mathrm{j}n\Omega}$为幅度恒为 1 的特殊复变量，将其推广至具有任意幅度和相位的新变量，称为 z，我们即可得到 z 变换的定义。

z 变换和拉普拉斯变换具有许多相同的性质，同时 z 变换和拉普拉斯变换在滤波器设计和分析中起着类似的作用。也就是说，z 变换对于数字滤波器的作用如同拉普拉斯变换对于模拟滤波器的作用。

z 变换的推导可由连续时间信号的拉普拉斯变换演变而来。对连续时间信号以周期 T 进行冲激抽样得到的连续时间信号 $x_s(t)$ 如下：

$$x_s(t)=\sum_{n=-\infty}^{\infty}x(t)\delta(t-nT)=\sum_{n=-\infty}^{\infty}x(nT)\delta(t-nT) \tag{3.59}$$

对式(3.59)两边同时进行拉普拉斯变换，得到

$$X_s(s)=\sum_{n=-\infty}^{\infty}x(nT)\mathrm{e}^{-nTs} \tag{3.60}$$

替换 $\mathrm{e}^{Ts}=z$，并利用 $x[n]$代替 $x(nT)$，重写式(3.60)为

$$X(z)=\sum_{n=-\infty}^{\infty}x[n]z^{-n}$$

在本节后面可以看到，z 变换可以通过离散信号的 DTFT 来推导。

3.7.1 z 变换定义

现在定义直接 z 变换和 z 反变换。通常去掉“直接”，简称直接 z 变换为 z 变换。通过该定义，离散时间信号 $x[n]$的 z 变换由累加给出

$$\mathrm{Z}[x[n]]=X(z)=\sum_{n=-\infty}^{\infty}x[n]z^{-n} \tag{3.61}$$

其中，Z[·]表示 z 变换。展开 z 变换可得

$$X(z)=\cdots+x[-2]z^2+x[-1]z+x[0]+x[1]z^{-1}+x[2]z^{-2}+\cdots$$

通常，z 为复数 $z=re^{j\Omega}$。(回顾拉普拉斯变换中变量 s 也是复数，$s=(\sigma+\mathrm{j}\omega)$，根据 $\mathrm{e}^{Ts}=z$ 可得 $r=\mathrm{e}^{\sigma T}$，$\Omega=\omega T$。)

式(3.61)定义的 z 变换称为双边 z 变换。z 反变换为

$$\mathrm{Z}^{-1}[X(z)]=x[n]=\frac{1}{2\pi\mathrm{j}}\int_{\Gamma}X(z)z^{n-1}\mathrm{d}z \tag{3.62}$$

其中，Z^{-1}[·]表示 z 反变换，Γ 是 z 平面上特定的逆时针闭合路径，式(3.62)称为复反演积分。由于计算该积分比较困难，因此很少直接利用它来计算 z 反变换，如其他变换同

样处理，一般利用表格进行求解。

式(3.61)和式(3.62)称为双边 z 变换对，下面将修改式(3.61)以获得实际应用中所需的 z 变换形式。首先将式(3.61)表示为

$$\mathrm{Z}[x[n]]=X(z)=\sum_{n=-\infty}^{-1}x[n]z^{-n}+\sum_{n=0}^{\infty}x[n]z^{-n}\tag{3.63}$$

然后定义 $x[n]=0$，当 $n<0$ 时。因此式(3.63)中的第一项累加为零，得到的变换称为单边 z 变换，其幂级数形式为

$$\mathrm{Z}_u[x[n]]=X_u(z)=\sum_{n=0}^{\infty}x[n]z^{-n}\tag{3.64}$$

$\mathrm{Z_u}[\cdot]$代表单边 z 变换，我们将这种单边 z 变换视为双边 z 变换的特例。

通常可能认为双边 z 变换适用于所有情况，无需单边 z 变换。然而，单边 z 变换提供了一种处理初始条件的简单方法，并且在求解因果且具有初始条件的差分方程中起到重要作用。除非特别指出，本书中都将采用双边 z 变换。"z 变换"意味着双边 z 变换，利用单边 z 变换时会使用术语"单边 z 变换"进行讨论。

双边 z 变换和单边 z 变换的 z 反变换(见式(3.62)是相同的，因此如用 $X_u(z)$代替 $X(z)$，式(3.62)则同样给出了单边 z 反变换。另外，单边 z 变换的 z 反变换给出了所有时间点的 $x[n]$，特别地，给出了 $x[n]=0$，$n<0$。如果 $x[n]$是可以进行 z 变换的(即(3.61)中的累加存在)，计算式(3.61)将得到函数 $X(z)$。计算 $X(z)$的反变换利用式(3.62)的复反演积分，可以产生 $x[n]$。将这种关系表示为

$$x[n]\overset{\mathrm{Z}}{\longleftrightarrow}X(z)\tag{3.65}$$

3.7.2 z 变换

本节将通过简单的应用介绍基于 z 变换的系统分析。首先，给出两个简单序列的 z 变换，接下来利用 z 变换求解一阶数字滤波器的阶跃响应。

两个 z 变换 在讨论这些例子之前，首先考虑下列收敛的幂级数：

$$\sum_{n=0}^{\infty}b^n=1+b+b^2+\cdots=\frac{1}{1-b};\quad |b|<1$$

随后将说明该级数在表示某些闭合形式的 z 变换(而不是以级数形式)时十分重要。这里更倾向于使用闭合形式的 z 变换，原因在于可使变换结果更加简洁。

回顾离散时间单位阶跃函数 $u[n-n_0]$：

$$u[n-n_0]=\begin{cases}1;&n\geqslant n_0\\0,&n<n_0\end{cases}\tag{3.66}$$

下面例子中将使用该离散时间单位阶跃函数。

例 3-18 单位阶跃函数的 z 变换

推导阶跃发生在 $n=0$ 的单位阶跃函数 z 变换，由式(3.63)和式(3.66)，可得

$$\mathrm{Z}[u[n]]=\sum_{n=-\infty}^{\infty}u[n]z^{-n}=\sum_{n=0}^{\infty}(1)z^{-n}=1+z^{-1}+z^{-2}+\cdots$$

令级数中 $a=z^{-1}$，得到闭合形式的 z 变换

$$\sum_{n=0}^{\infty}z^{-n}=\frac{1}{1-z^{-1}}=\frac{z}{z-1}$$

因此，单位阶跃函数的 z 变换仅在$|z^{-1}|<1$处存在，也就是说，对单位圆外的所有 z，有 z 变换对

$$u[n]\overset{\mathrm{Z}}{\longleftrightarrow}\frac{1}{1-z^{-1}};\quad |z^{-1}|<1$$

◀

单位阶跃函数的 z 变换说明了标注的一个问题。式(3.61)中 z 变换的定义得到的是 z^{-1} 的函数，然而我们通常习惯用正指数表达的变量。一些作者将所有的 z 变换用负指数表示。这里选择转换为 z 的正幂来表示 $X(z)$，正是由于该种标注更为普遍而且不易出错。因此，例 3-18 将单位阶跃函数的 z 变换表示为

$$u[n] \overset{Z}{\longleftrightarrow} \frac{z}{z-1}; \quad |z|>1 \tag{3.67}$$

例 3-19 指数函数的 z 变换

求解指数函数 $x[n]=a^n u[n]$ 的 z 变换。a^n 是离散时间指数函数，由(3.64)可得

$$X(z)=\sum_{n=0}^{\infty} a^n z^{-n}=1+az^{-1}+a^2z^{-2}+\cdots=1+az^{-1}+(az^{-1})^2+\cdots$$

以闭合形式给出 z 变换为

$$Z[a^n u[n]]=\sum_{n=0}^{\infty}(az^{-1})^n=\frac{1}{1-az^{-1}}=\frac{z}{z-a}; \quad |z|>|a|$$

同时得到 z 变换对

$$a^n u[n] \overset{Z}{\longleftrightarrow} \frac{z}{z-a}; \quad |z|>|a|$$ ◀

如同例 3-18 和例 3-19 所示，单位阶跃函数 $u[n]$ 和指数函数 $a^n u[n]$ 的 z 变换的存在都有条件。任意函数的 z 变换存在都是有条件的，以使此无限级数

$$Z[x[n]]=X(z)=\sum_{n=-\infty}^{\infty} x[n]z^{-n}$$

收敛。

在式(3.62)反变换的积分路径 Γ 由 $X(z)$ 的收敛域(Region of Convergence，ROC)确定。在利用 z 变换表求解反变换时，需要知道 $X(z)$ 的收敛域。

例 3-19 给出了因果函数的 $a^n u[n]$ 的 z 变换对为

$$a^n u[n] \overset{F_b}{\longleftrightarrow} \frac{z}{z-a}; \quad |z|>|a| \tag{3.68}$$

可见，z 变换需标明其 ROC。

式(3.68)的指数函数如图 3.22a 所示，同时给出了其 z 变换的 ROC，其中 a 为实数。为了说明 ROC 的重要性，下面推导 $-a^n u[-n-1]$ 的双边 z 变换，该指数函数如图 3.22b 所示，a 为实数。由式(3.61)

$$\begin{aligned} Z[-a^n u[-n-1]] &= \sum_{n=-\infty}^{\infty} -a^n u[-n-1]z^{-n}=\sum_{n=-\infty}^{0} -a^n z^{-n} \\ &= -(a^{-1}z+a^{-2}z^2+a^{-3}z^3+\cdots)=\sum_{n=1}^{\infty} -(a^{-1}z)^n \end{aligned} \tag{3.69}$$

因为当 $n\geqslant 0$ 时，$u[-n-1]$ 为零，根据收敛幂级数

$$\sum_{n=k}^{\infty} b^n=\frac{b^k}{1-b}; \quad |b|<1 \tag{3.70}$$

根据(3.69)，并令 $b=a^{-1}z$，$k=1$，得到 z 变换

$$Z[-a^n u[-n-1]]=\frac{-a^{-1}z}{1-a^{-1}z}=\frac{z}{z-a}; \quad |a^{-1}z|<1 \tag{3.71}$$

该变换的 ROC 可以表示为 $|z|<|a|$，如图 3.22b 所示。

对比式(3.68)和式(3.71)的双边 z 变换对：

$$a^n u[n] \overset{F_b}{\longleftrightarrow} \frac{z}{z-a}; \quad |z|>|a|$$

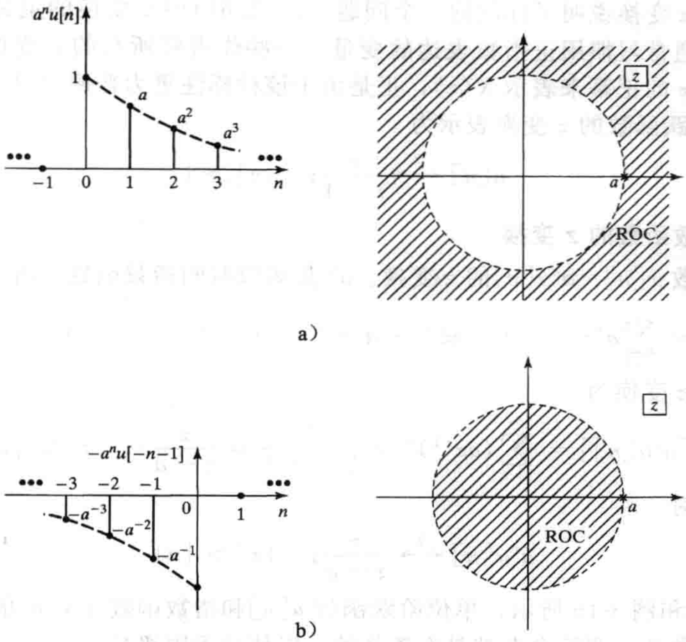


图 3.22　指数函数及其收敛域

$$-a^n u[-n-1] \overset{F_b}{\longleftrightarrow} \frac{z}{z-a}; \quad |z|<|a|$$

可以看出，两个 z 变换结果相同，只是 ROC 不同。因此，在求解反变换之前必须已知 z 变换对应的 ROC。

例 3-20　时间翻转单位阶跃函数的 z 变换

考虑单位阶跃函数 $u[n]$，由(3.67)可得 z 变换对

$$u[n] \overset{z}{\longleftrightarrow} \frac{z}{z-1}; \quad |z|>1$$

对于时间翻转的单位阶跃函数 $-u[-n-1]$，由式(3.71)，令 $a=1$ 得到变换对

$$Z[-u[-n-1]] = \frac{z}{z-1}; \quad |z|<1 \tag{3.72}$$

两个单位阶跃函数的 z 变换是相同的，但 ROC 不同。◀

关于信号和双边 z 变换，给出 4 个有用的定义：

1. 如果对于 $n<n_0$，$x[n]=0$，称函数 $x[n]$是右边函数，其中，n_0 可为零、正整数或负整数。如，$u[n+10]$是右边函数，因为

$$u[n+10] = \begin{cases} 0, & n<-10 \\ 1, & n \geqslant -10 \end{cases}$$

2. 如果对于 $n>n_0$，$x[n]=0$，称函数 $x[n]$是左边函数，其中，n_0 可为零、正整数或负整数。如，$u[-n+10]$是右边函数，因为

$$u[-n+10] = \begin{cases} 1, & n \leqslant 10 \\ 0, & n>10 \end{cases}$$

3. 如果函数 $x[n]$既不是左边函数又不是右边函数，则 $x[n]$是双边函数。如，$\cos(n)$是双边的。

4. 如果函数既是右边函数又是左边函数，则该函数持续时间是有限的。如，$(u[n]-u[n-10])$是有限长函数。

当处理双边 z 变换时，以上 4 个定义十分重要。

3.7.3　离散时间函数的 z 变换

表 3.9 给出了常见离散信号的 z 变换。下面给出使用该表的过程。

根据函数 $x[n]$，定义两个函数 $x^+[n]$ 和 $x^-[n]$：

$$x^+[n] = x[n]u[n],\quad x^-[n] = x[n]u[-n-1]$$

因此，$x^+[n]$ 是右边函数，$x^-[n]$ 是左边函数。如果 $X(z)$ 存在，则可将 $x[n]$ 表示为

$$x[n] = x^+[n] + x^-[n] \Rightarrow X(z) = X^+(z) + X^-(z) \tag{3.73}$$

除了表 3.9 中的最后一项，表 3.9 中的每个变换都可以表述为 $X^+(z)$ 或者 $X^-(z)$。现在讨论利用表 3.9 的例子。

表 3.9　z 变换对

离散时间序列	z 变换	ROC
$\delta[n]$	1	z 平面
$\delta[n-m]$	z^{-m}	z 平面，除了 $\begin{cases} z=0, & m>0 \\ z=\infty, & m<0 \end{cases}$
$u[n]$	$\dfrac{z}{z-1}$	$\lvert z\rvert>1$
$-u[-n-1]$	$\dfrac{z}{z-1}$	$\lvert z\rvert<1$
$a^n u[n]$	$\dfrac{z}{z-a}$	$\lvert z\rvert>\lvert a\rvert$
$-a^n u[-n-1]$	$\dfrac{z}{z-a}$	$\lvert z\rvert<\lvert a\rvert$
$na^n u[n]$	$\dfrac{az}{(z-a)^2}$	$\lvert z\rvert>\lvert a\rvert$
$-na^n u[-n-1]$	$\dfrac{az}{(z-a)^2}$	$\lvert z\rvert<\lvert a\rvert$
$\cos(\Omega_0 n)u[n]$	$\dfrac{z(z-\cos(\Omega_0))}{z^2-2\cos(\Omega_0)z+1}$	$\lvert z\rvert>1$
$\sin(\Omega_0 n)u[n]$	$\dfrac{\sin(\Omega_0)z}{z^2-2\cos(\Omega_0)z+1}$	$\lvert z\rvert>1$
$r^n\cos(\Omega_0 n)u[n]$	$\dfrac{z(z-r\cos(\Omega_0))}{z^2+2r\cos(\Omega_0)z+r^2}$	$\lvert z\rvert>r$
$r^n\sin(\Omega_0 n)u[n]$	$\dfrac{r\sin(\Omega_0)z}{z^2+2r\cos(\Omega_0)z+r^2}$	$\lvert z\rvert>r$
$x[n]=\begin{cases} a^n, & 0\leqslant n\leqslant N-1 \\ 0, & \text{其他} \end{cases}$	$\dfrac{z^N-a^N}{z^N-az^{N-1}}$	$\lvert z\rvert>0$

例 3-21　指数函数的 z 变换

利用(3.73)和表 3.9，可以求解一个双边函数 $x[n]=a^{|n|}$ 的双边 z 变换。将 $x[n]$ 表示为

$$\begin{aligned} x[n] &= a^{|n|} = a^{|n|}[u[n]+u[-n-1]] \\ &= a^n u[n] + a^{-n}u[-n-1] = x^+[n] + x^-[n] \end{aligned}$$

由表 3.9

$$X^+(z) = Z[x^+[n]] = Z[a^n u[n]] = \frac{z}{z-a};\quad |z| > |a|$$

对于左边函数

$$X^-(z)=Z[x^-[n]]=Z[a^{-n}u[-n-1]]=\frac{-z}{z-1/a};\quad |z|<1/|a|$$

因此，双边函数 $x[n]=a^{|n|}$ 的 z 变换为

$$Z[a^{|n|}]=\frac{z}{z-a}-\frac{z}{z-1/a};\quad |a|<|z|<1/|a|$$ ◀

仅当 $|a|<1$ 时该变换存在。

3.7.4 收敛域

在讨论 z 反变换之前，这里将进一步探究收敛域。首先考虑一个特殊的例子，假设函数 $x[n]$ 在 n 等于零和 1 之外全为零，因此 $x[n]$ 持续时间有限，且

$$X(z)=x[0]+x[1]z^{-1} \tag{3.74}$$

除了在 $z=0$ 处外，$X(z)$ 在 z 平面其他任意点都存在，所以 ROC 为除去原点之外的整个 z 平面。

现在考虑 $x[n]$ 只在 $n=-1$ 和 $n=0$ 处有非零值的情况，同样地，$x[n]$ 持续时间有限，因此

$$X(z)=x[-1]z+x[0] \tag{3.75}$$

在有限平面内，函数在任意点都存在，因此其 ROC 是有限平面。

接下来考虑 $x[n]$ 是持续时间有限的任意序列。通常，$X(z)$ 只在原点(如在式(3.74)中)或者在无穷点(如在式(3.75)中)处有极点。因此，得出结论，有限序列的 z 变换，除了可能在原点处不存在，在整个有限 z 平面的任意点都存在。

由本节的讨论可见，ROC 范围由中心在原点的圆确定，可能不包括原点。因此，z 变换的 ROC 为以下 4 种形式之一：

1. 中心在原点的圆的外部(右边函数)；
2. 中心在原点的圆的内部，可能除去原点(左边函数)；
3. 中心在原点的环形区域(双边函数)；
4. 整个有限平面，可能除去原点(持续时间有限函数)。

上述 ROC 都是开区域，即 ROC 的边界不包含在收敛区域。

最后得到一个性质：由于 z 变换在其极点处是不收敛的，因此函数 $X(z)$ 的所有极点都在其 ROC 的外部。ROC 的边界始终包含极点，可能还有其他极点位于 ROC 的外部。

3.7.5 *z* 反变换

为了确定双边 z 反变换，首先求解 $X(z)$ 的部分分式展开式，然后将 $X(z)$ 表示为两个函数之和，如式(3.73)：

$$X(z)=X^+(z)+X^-(z)\Rightarrow x[n]=x^+[n]+x^-[n] \tag{3.76}$$

这里 $X^+(z)$ 表示右边函数，包含其极点位于 ROC 内边界之里的展开项，$X^-(z)$ 表示左边函数，包含其极点位于 ROC 外边界之外的展开项。然后直接由表 3.9 得到反变换。

下面给出三个例子进行说明，这三个例子中的双边 z 变换均相同而 ROC 不同。示例中使用的双边 z 变换为

$$X(z)=\frac{2z^2-0.75z}{(z-0.25)(z-0.5)}=\frac{z}{z-0.25}+\frac{z}{z-0.5}$$

例 3-22 ***z* 反变换**

首先考虑函数：

$$X(z)=\frac{z}{z-0.25}+\frac{z}{z-0.5};\quad |z|>0.5$$

极点和 ROC 如图 3.23 所示，可知 $x[n]$是右边序列，由表 3.9 得

$$x[n]=Z^{-1}[X(z)]=[0.25^n+0.5^n]u[n]$$

◀

例 3-23 **例 3-22 续**

接下来考虑函数：

$$X(z)=\frac{z}{z-0.25}+\frac{z}{z-0.5};\quad |z|<0.25$$

极点和 ROC 如图 3.24 所示，可知，$x[n]$是左边序列，由表 3.9 得

$$x[n]=Z^{-1}[X(z)]=[-0.25^n-0.5^n]u[-n-1]$$

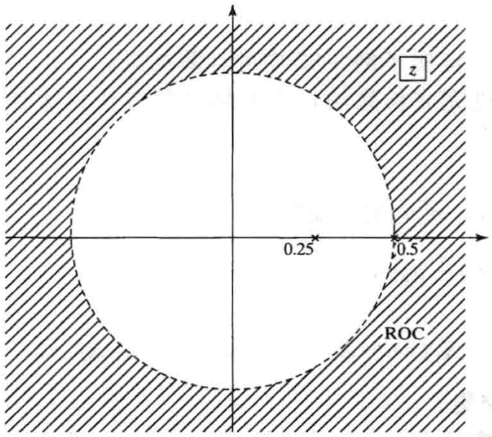


图 3.23 例 3-22 的 ROC

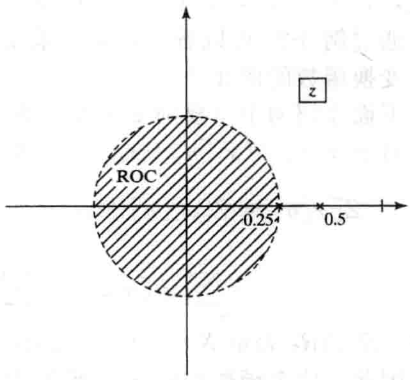


图 3.24 例 3-23 的 ROC

◀

例 3-24 **例 3-23 续**

最后考虑函数

$$X(z)=\frac{z}{z-0.25}+\frac{z}{z-0.5};\quad 0.25<|z|<0.5$$

极点和 ROC 如图 3.25 所示。

函数$z/(z-0.25)$是右边函数的 z 变换，函数 $z/(z-0.5)$是左边函数的 z 变换。由表 3.9 得

$$x[n]=Z^{-1}[X(z)]=0.25^nu[n]-0.5^nu[n-1]$$

◀

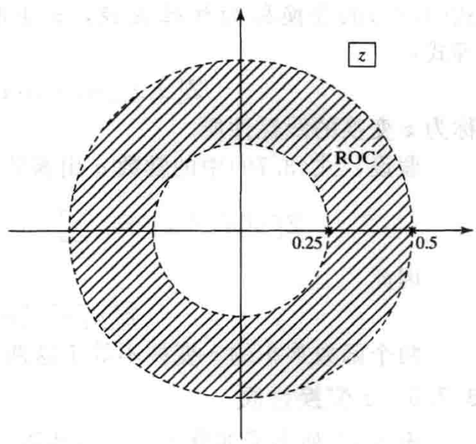


图 3.25 例 3-24 的收敛域

下面介绍两个重要的 z 变换性质。

第三种计算 z 反变换的方法就是通过长除法将有理分式的 z 变换展开成幂级数。具体展开方法前面已有说明，即利用 z 变换表示式中的分子除以其分母，得到的结果是幂级数

$$X(z)=\frac{N(z)}{D(z)}=x_0+x_1z^{-1}+x_2z^{-2}+\cdots \quad (3.77)$$

对比该级数与双边 z 变换的定义，可以看到式(3.64)中的 $x[n]$等于式(3.77)中的 x_n。下面通过例子说明该过程。

例 3-25 **利用长除法计算 z 反变换**

现用长除法验证例 3-22 的反变换。对于例 3-22，

$$X(z)=\frac{z}{z-0.25}+\frac{z}{z-0.5}=\frac{2z^2-0.75z}{(z-0.25)(z-0.5)}=\frac{2z^2-0.75z}{z^2-0.75z+0.125}$$

分子除以分母得到

$$\begin{array}{r} 2+0.75z^{-1}+0.3125z^{-2}\cdots \\ z^2-0.75z+0.125\ \overline{)\,2z^2-0.75z\qquad\qquad\qquad} \\ \underline{2z^2-1.5z+0.25\qquad\qquad} \\ 0.75z-0.25\qquad\qquad \\ \underline{0.75z-0.5625+0.09375z^{-1}} \\ 0.3125-0.09375z^{-1} \\ \vdots \end{array}$$

因此，$x[0]=2.0$，$x[1]=0.75$，$x[2]=0.3125$，证实了例3-22中的计算结果。◀

通过例3-25可以看出，除非采用数字计算机完成长除法，否则幂级数方法只适合计算 z 变换函数的前几个值。

下面介绍两个重要的 z 变换性质。

对于 $x[n]=(x_1[n]+x_2[n])$，根据式(3.61)，$x[n]$的 z 变换为

$$\begin{aligned} Z[x[n]]&=Z[x_1[n]+x_2[n]]=\sum_{n=-\infty}^{\infty}[x_1[n]+x_2[n]]z^{-n} \\ &=\sum_{n=-\infty}^{\infty}x_1[n]z^{-n}+\sum_{n=-\infty}^{\infty}x_2[n]z^{-n}=X_1(z)+X_2(z),\mathrm{ROC}=R_1\cap R_2 \end{aligned} \tag{3.78}$$

其中，$R_1\cap R_2$ 表示 $X_1(z)$和 $X_2(z)$的收敛域的交集。

因此，两个函数和的 z 变换等于函数 z 变换之和。(假设这两个函数的 z 变换均存在。)通过将前面推导中的 $x_2[n]$替换为 $x_3[n]+x_4[n]$并以此类推，该性质可扩展到任意多个函数和的情况。

为推导 z 变换的第二个性质，研究 $ax[n]$的 z 变换，其中，a 为常数：

$$Z[ax[n]]=\sum_{n=0}^{\infty}ax[n]z^{-n}=a\sum_{n=0}^{\infty}x[n]z^{-n}=aX(z) \tag{3.79}$$

因此，函数乘以常数 a 的 z 变换等于常数乘以此函数的 z 变换。具有性质式(3.78)和式(3.79)的变换称为线性变换，z 变换是一个线性变换。这两个性质通常表示为一个等式：

$$Z[a_1x_1[n]+a_2x_2[n]]=a_1X_1(z)+a_2X_2(z) \tag{3.80}$$

称为 z 变换的线性性质。

假设，式(3.79)中的常数 a 用函数 $y[n]$代替，则

$$Z[y[n]x[n]]=\sum_{n=-\infty}^{\infty}y[n]x[n]z^{-n}\neq\sum_{n=-\infty}^{\infty}y[n]z^{-n}\sum_{n=-\infty}^{\infty}x[n]z^{-n}$$

因此，

$$Z[y[n]x[n]]\neq Z[y[n]]Z[x[n]] \tag{3.81}$$

两个函数乘积的 z 变换不等于这两个函数各自 z 变换的乘积。

3.7.6 z 变换性质

表3.10列出了部分 z 变换的性质。

例3-26 初值和终值性质

这里通过举例来说明初值和终值性质。单位阶跃函数 $u[n]$的 z 变换为

$$Z[u[n]]=\frac{z}{z-1};\quad |z|>1$$

$u[n]$是 $n_0=0$ 的右边函数，可采用单边 z 变换，由表3.10中的初值性质得，

$$x[0] = \lim_{z\to\infty} \frac{z}{z-1} = \lim_{z\to\infty} \frac{1}{1-1/z} = 1$$

已知 $u[n]$的终值是存在的，因此由表 3.10 中的终值性质得，

$$x[\infty] = \lim_{z\to 1}(z-1)X(z) = \lim_{z\to 1}(z-1)\frac{z}{z-1} = \lim_{z\to 1} z = 1$$

可见这两个值都是正确的。 ◀

表 3.10　z 变换的性质

性质	变换对	收敛域
z 变换表示	$x[n] \overset{Z}{\longleftrightarrow} X(z)$	R_x
	$x_1[n] \overset{Z}{\longleftrightarrow} X_1(z)$	R_1
	$x_2[n] \overset{Z}{\longleftrightarrow} X_2(z)$	R_2
线性	$ax_1[n]+bx_2[n] \overset{Z}{\longleftrightarrow} aX_1(z)+bX_2(z)$	至少 $R_1 \cap R_2$
时移	$x[n-k] \overset{Z}{\longleftrightarrow} z^{-k}X(z)$	R_x 可能比原来的收敛域增加或减少
时间反转	$x[-n] \overset{Z}{\longleftrightarrow} X(z^{-1})$	R_x^{-1}
频移	$e^{j\Omega_0 n}x[n] \overset{Z}{\longleftrightarrow} X(e^{-j\Omega_0 n}z)R_x$	R_x
卷积	$x_1[n] * x_2[n] \overset{Z}{\longleftrightarrow} X_1(z)X_2(z)$	至少 $R_1 \cap R_2$
z 域微分	$nx[n] \overset{Z}{\longleftrightarrow} -z\dfrac{dX(z)}{dz}$	R_x
初值	$x[0]=\lim_{z\to\infty} X(z)$，如果 $n<0$，$x[n]=0$	
终值	$\lim_{n\to\infty} x[n]=\lim_{z\to 1}(z-1)X(z)$，如果 $\lim_{n\to\infty} x[n]$存在	

例 3-27　例 3-26 续

现在考虑正弦函数 $\sin(\pi n/2)u[n]$。由表 3.9 得，

$$Z[\sin(\pi n/2)] = \frac{z\sin(\pi/2)}{z^2 - 2z\cos(\pi/2) + 1} = \frac{z}{z^2+1}; \quad |z| > 1$$

由表 3.10 中的初值性质得，

$$x[0] = \lim_{z\to\infty} X(z) = \lim_{z\to\infty} \frac{z}{z^2+1} = 0$$

由终值性质得，

$$x[\infty] = \lim_{z\to 1}(z-1)\frac{z}{z^2+1} = 0$$

结果是错误的，原因在于 $\sin(\pi n/2)u[n]$会继续震荡，因此没有终值。 ◀

由此可见，利用 $X(z)$求解 $x[n]$终值时，首先是 $x[n]$的终值存在。

例 3-28　时移性质

描述系统的差分方程为

$$y[n] = 0.9y[n-1] - 0.2y[n-2] + 0.3x[n-1]$$

且 $y[-1]=y[-2]=0$，$x[-1]=0$。

利用线性性质和时移性质求解此方程的 z 变换

$$Y(z) = 0.9z^{-1}Y(z) - 0.2z^{-2}Y(z) + 0.3z^{-1}X(z)$$

将含 $Y(z)$的项移至方程左侧并整理得

$$(1 - 0.9z^{-1} + 0.2z^{-2})Y(z) = 0.3z^{-1}X(z)$$

求解 $Y(z)$并用 z 的正指数形式表示

$$Y(z) = \frac{0.3z}{z^2 - 0.9z + 0.2}X(z)$$

下一节将推导其他性质。

3.7.7 LTI 系统的应用

本节将讨论 z 变换在线性非时变(LTI)系统中的应用。首先讨论系统的传输函数，然后研究系统的某些性质。

传输函数 若描述某因果离散时间系统的数学模型为常系数的线性差分方程，则该离散时间系统是线性非时变系统。模型的通用方程为

$$\sum_{k=0}^{N} a_k y[n-k] = \sum_{k=0}^{M} b_k x[n-k] \tag{3.82}$$

其中 $x[n]$为输入信号，$y[n]$为输出信号，N 为离散系统的阶数，常数 a_k、b_k、N、M 为离散系统的参数。对于因果系统，$M \leqslant N$。为了方便，这里假设系统是因果的，并令 $M=N$。(如果 $M<N$，且系统为因果系统，右侧累加和中的某些系数必须为零。)

由表 3.10 中的时移性质，可得

$$Z[f[n-n_0]] = z^{-n_0} F(z) \tag{3.83}$$

因此，当 $M=N$ 时，式(3.82)的 z 变换为

$$\sum_{k=0}^{N} a_k z^{-k} Y(z) = \sum_{k=0}^{N} b_k z^{-k} X(z) \tag{3.84}$$

其中，$n<0$ 时，$x[n]=y[n]=0$。将式(3.84)展开为

$$\begin{aligned} &[a_0 + a_1 z^{-1} + \cdots + a_{N-1} z^{-N+1} + a_N z^{-N}] Y(z) \\ &= [b_0 + b_1 z^{-1} + \cdots + b_{N-1} z^{-N+1} + b_N z^{-N}] X(z) \end{aligned} \tag{3.85}$$

定义离散系统的传输函数 $H(z)$为$Y(z)/X(z)$，因此，由式(3.85)可得，

$$\begin{aligned} H(z) &= \frac{Y(z)}{X(z)} = \frac{b_0 + b_1 z^{-1} + \cdots + b_{N-1} z^{-N+1} + b_N z^{-N}}{a_0 + a_1 z^{-1} + \cdots + a_{N-1} z^{-N+1} + a_N z^{-N}} \\ &= \frac{b_0 z^N + b_1 z^{N-1} + \cdots + b_{N-1} z + b_N}{a_0 z^N + a_1 z^{N-1} + \cdots + a_{N-1} z + a_N} \end{aligned} \tag{3.86}$$

其中最后一步是将变量 z 以正指数形式表示。下面举例说明。 ◀

例 3-29 由系统的差分方程求解传输函数

根据例 3-28 中系统的差分方程和 z 变换方程。系统传输函数定义为输出信号 z 变换与输入信号 z 变换之比：

$$H(z) = \frac{Y(z)}{X(z)} = \frac{0.3z}{z^2 - 0.9z + 0.2}$$ ◀

例 3-30 由系统框图求解传输函数

如图 3.26a 所示的 α 滤波器，描述系统的差分方程为

$$y[n] - (1-\alpha) y[n-1] = \alpha x[n]$$

此例中，令 $\alpha=0.1$，则有

$$y[n] - 0.9y[n-1] = 0.1x[n]$$

将方程两边进行 z 变换可得

$$(1-0.9z^{-1}) Y(z) = 0.1X(z)$$

因此系统的传输函数为

$$H(z) = \frac{Y(z)}{X(z)} = \frac{0.1}{1-0.9z^{-1}} = \frac{0.1z}{z-0.9}$$ ◀

注意，也可以直接由式(3.84)与式(3.86)得到例 3-29 和例 3-30 中的系统传输函数。

通常利用传输函数 $H(z)$或差分方程来描述滤波器。例 3-30 中图 3.26 所示的 α 滤波器的框图可以采用 4 种形式中的任意一种。传输函数中用 z^{-1}表示时延基于以下性质

$$Z[y[n-1]] = z^{-1} Z[y[n]] = z^{-1} Y(z)$$

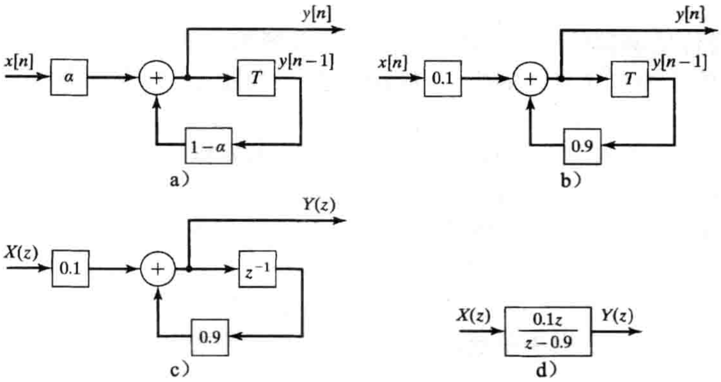


图 3.26 α滤波器框图

因果性 下面讨论 LTI 因果系统。对于传输函数

$$H(z)=\frac{z^2+0.4z+0.9}{z-0.6}=z+\frac{z+0.9}{z-0.6} \tag{3.87}$$

其中最后一个函数表示式可通过 $H(z)$的分子除以分母得到。该系统是非因果的，因为该系统可以表示为一个物理可实现系统并联一个单位时间前移系统。由于 $H(z)$分子的阶数比分母高，因此出现了单位时间前移。

根据以前的讨论可知，对于一个因果系统，当变量 z 以正指数形式表示时，式(3.86)中传输函数$H(z)$分子的阶数不能高于分母的阶数。如果传输函数采用式(3.86)中的负指数形式表示，也就是，

$$H(z)=\frac{Y(z)}{X(z)}=\frac{b_0+b_1z^{-1}+\cdots+b_{N-1}z^{-N+1}+b_Nz^{-N}}{a_0+a_1z^{-1}+\cdots+a_{N-1}z^{-N+1}+a_Nz^{-N}} \tag{3.88}$$

如果 $a_0\neq0$，此系统是因果系统。

稳定性 现在将因果系统有界输入有界输出(BIBO)的稳定性与系统传输函数联系起来。首先回顾一下 BIBO 稳定性的定义。

BIBO 稳定性 如果系统对于任意有界的输入，其输出均有界，则该系统为稳定的。

对于 N 阶离散时间 LTI 因果系统，传输函数可以表示为

$$[(3.86)]\qquad H(z)=\frac{Y(z)}{X(z)}=\frac{b_0z^N+b_1z^{N-1}+\cdots+b_{N-1}z+b_N}{a_0z^N+a_1z^{N-1}+\cdots+a_{N-1}z+a_N}$$

其中 $a_0\neq0$。该传输函数的分母可通过因式分解为

$$a_0z^N+a_1z^{N-1}+\cdots+a_{N-1}z+a_N=a_0(z-p_1)(z-p_2)\cdots(z-p_N) \tag{3.89}$$

这些多项式的零点为传输函数的极点。根据定义，极点是使得 $H(z)$不收敛的 z 值点。

首先假设 $H(z)$没有重极点，利用部分分式法表示式(3.86)中的输出 $Y(z)$为

$$\begin{aligned}Y(z)=H(z)X(z)&=\frac{b_0z^N+b_1z^{N-1}+\cdots+b_{N-1}z+b_N}{a_0(z-p_1)(z-p_2)\cdots(z-p_N)}X(z)\\&=\frac{k_1z}{z-p_1}+\frac{k_2z}{z-p_2}+\cdots+\frac{k_Nz}{z-p_N}+Y_x(z)\end{aligned} \tag{3.90}$$

其中，部分分式法中的 $Y_x(z)$来源于输入信号 $X(z)$的极点所对应的各项之和，因此 $Y_x(z)$对应系统的强迫响应。

在式(3.90)的部分分式中，假设 $H(z)$分子的阶数比分母的阶数低。如果多项式分子的阶数等于或者大于多项式分母的阶数，部分分式就包含如式(3.87)所示的附加项。

式(3.90)的反变换为

$$y[n]=k_1p_1^n+k_2p_2^n+\cdots+k_Np_N^n+y_x[n]=y_n[n]+y_x[n] \tag{3.91}$$

$y_n[n]$项来自传输函数的极点，因而$y_n[n]$是系统的自由响应。无论输入信号$x[n]$是何种形式，系统输出均包含自由响应。自由响应中的每项p_i^n称为系统的响应形式。

如果输入信号$x[n]$有界，则强迫响应$y_x[n]$有界，因为$y_x[n]$与$x[n]$具有相同的函数形式。($Y_x(z)$与$X(z)$拥有相同的极点。)因此，一个无界输出必定是至少存在一个自由响应项$k_ip_i^n$为无界所导致的。只要至少一个极点的模$|p_i|$大于1，这种无界性就会产生。

根据定义，系统特征方程是将传输函数的分母多项式置零。因此，特征方程是式(3.86)中的分母为零，即

$$a_0z^N+a_1z^{N-1}+\cdots+a_{N-1}z+a_N=a_0(z-p_1)(z-p_2)\cdots(z-p_n)=0 \tag{3.92}$$

此为系统的特征方程。对于因果系统，如果特征方程的根都在z平面的单位圆内，则系统是稳定的。对于$H(z)$有重极点的情况，上述结论依然适用。下面举例说明系统的稳定性。

例 3-31 由传输函数判定系统的稳定性

考虑例3-28和例3-29中的系统。在例3-29中，该因果系统的传输函数为

$$H(z)=\frac{Y(z)}{X(z)}=\frac{0.3z}{z^2-0.9z+0.2}$$

将特征多项式置为零，得到系统的特征方程：

$$z^2-0.9z+0.2=(z-0.4)(z-0.5)=0$$

特征方程的根即传输函数的极点决定了系统的稳定性。系统的响应形式为

$$(0.4)^n \text{ 和 } (0.5)^n$$

由于该特征方程的两个根的模都小于1(对应的极点都在单位圆内)，因此该系统是稳定的。如果其中有一个极点在单位圆外，则系统不稳定。 ◀

可逆性

逆系统 系统$H(z)$的逆系统$H_i(z)$是另一个系统，当其与$H(z)$级联时，得到一个传输函数为1的系统。

因此，由定义可得$H_i(z)$为

$$H(z)H_i(z)=1\Rightarrow H_i(z)=\frac{1}{H(z)} \tag{3.93}$$

下面讨论逆系统的特征。因果系统的传输函数$H(z)$可以表示为

$$H(z)=\frac{b_0z^M+b_1z^{M-1}+\cdots+b_M}{a_0z^N+a_1z^{N-1}+\cdots+a_N} \tag{3.94}$$

其中，分子的阶数小于或者等于分母的阶数($M\leqslant N$)。因此，其逆系统的传输函数为

$$H_i(z)=\frac{a_0z^N+a_1z^{N-1}+\cdots+a_N}{b_0z^M+b_1z^{M-1}+\cdots+b_M} \tag{3.95}$$

如果$N\leqslant M$，这个系统是因果的。因此，为保证该系统及其逆系统均为因果，则有$M=N$。

接下来讨论系统稳定性。下面讨论系统的稳定性。为了使式(3.94)表示的因果系统稳定，传输函数的所有极点必须都在z平面的单位圆内。为了使逆系统(3.95)稳定，$H_i(z)$的极点(即$H(z)$的零点)必须都在z平面的单位圆内。为了使图3.27中的系统$H(z)$稳定，则要求其所有的零点和极点均在z平面的单位圆内。

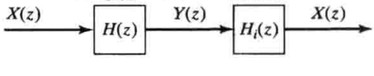


图 3.27 系统及其逆系统

3.7.8 离散时间傅里叶变换与z变换的关系

现在将DTFT与z变换联系起来。离散序列$x[n]$的双边z变换定义为

[(3.61)] $$Z[x[n]]=X(z)=\sum_{n=-\infty}^{\infty}x[n]z^{-n}$$

表3.9列出了部分离散时间函数的双边 z 变换。离散序列 $x[n]$ 的DTFT定义为

[(3.29)] $$X(\Omega)=\mathrm{DFT}\{X[n]\}=\sum_{n=-\infty}^{\infty}x[n]\mathrm{e}^{-\mathrm{j}n\Omega}$$

对比式(3.61)与式(3.29)可知，z 变换与DTFT的关系如下：

$$\mathrm{DTF}\{x[n]\}=Z[x[n]]\big|_{z=\mathrm{e}^{\mathrm{j}\Omega}} \tag{3.96}$$

由此方程可以看到，如果离散信号的DTFT存在，则可通过 z 变换求其DTFT。

这里无需证明，如果 $x[n]$ 是绝对可和的，则 $\mathrm{DTF}\{x[n]\}$ 存在[2]，即

$$\sum_{n=-\infty}^{\infty}|x[n]|<\infty \tag{3.97}$$

这个条件是充分条件。一些不满足此条件的离散信号确实存在DTFT。通常，这些离散信号的DTFT存在，且是以 Ω 为变量的冲激函数，将在以后讨论这些信号的变换。

在表3.9中给出了 z 变换的收敛域，这些收敛区域 $a<|z|<b$ 是环状的。为了利用该表求解DTFT，把 $z=\mathrm{e}^{\mathrm{j}\Omega}$ 代入式(3.96)，则收敛域变为

$$a<|z|<b\Rightarrow a<|\mathrm{e}^{\mathrm{j}\Omega}|<b$$

由于 $|\mathrm{e}^{\mathrm{j}\Omega}|=1$，对 $|\mathrm{e}^{\mathrm{j}\Omega}|$ 作图，其对应 z 平面上的单位圆。因此，表3.9中收敛域包含单位圆的 z 变换，可直接替换为函数对应的DTFT，该性质在图3.28中已给出说明。在图3.28a中，单位圆位于收敛域之内，则DTFT存在。在图3.28b中，单位圆位于收敛域之外，则DTFT不存在。下面将举例进行说明。

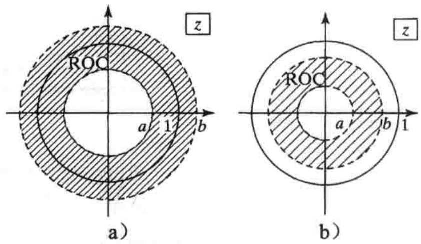


图3.28 收敛域

例3-32 利用 z 变换求解DTFT

求解函数 $x[n]=na^nu[n]$ 的DTFT。由表3.9得

$$Z[na^nu[n]]=\frac{az}{(z-a)^2};\quad |z|>|a|$$

因此，为了使得 $na^nu[n]$ 的DTFT存在，则需 $|\mathrm{e}^{\mathrm{j}\Omega}|=1>|a|$，因此可得DTFT为

$$\mathrm{DTF}\{na^nu[n]\}=\frac{a\mathrm{e}^{\mathrm{j}\Omega}}{(\mathrm{e}^{\mathrm{j}\Omega}-a)^2};\quad |a|<1$$

◀

计算频率响应 对于由 z 变换传输函数描述的离散系统，式(3.96)提供了计算其频率响应的方法。将传输函数中的 z 以 $\mathrm{e}^{\mathrm{j}\Omega}$ 代替，即可求解离散系统对应的频率响应。

例3-33 由 z 变换传输函数求解频率响应

对于例3-30中的 α 滤波器，抽样间隔 $T=0.01\mathrm{s}$。描述该滤波器的 z 变换传输函数为

$$H(z)=\frac{Y(z)}{X(z)}=\frac{0.1}{1-0.9z^{-1}}$$

为了计算频率响应，利用式(3.96)得到

$$H(\Omega)=H(z)\big|_{z=\mathrm{e}^{\mathrm{j}\omega T}}=\frac{0.1}{1-0.9\mathrm{e}^{-\mathrm{j}\Omega}}=\frac{0.1}{1-0.9\mathrm{e}^{-\mathrm{j}\Omega}}=\frac{0.1}{1-0.9\mathrm{e}^{-\mathrm{j}0.01\omega}},\quad \omega<100\pi \tag{3.98}$$

通过代入选定的 ω 值，由式(3.98)可计算出系统的频率响应。

采用 ω 的300个值，通过下列MATLAB程序计算并画出系统的幅度谱和相位谱。

```
w=[0:1:300]; % Est. frequency range in radians per second)
T=0.01; %( Sample rate: 100 samples per second)
W=w*T; %W is in radians per sample.
numH=[0.1]; denH=[1 -0.9];
HW=freqz(numH, denH, W);
magH=abs(HW); phaseH=angle(HW)*180/pi;
subplot(2,1,1); plot(w,magH),grid, %Frequencies plotted are % in rad/s.
subplot(2,1,2); plot(w,phaseH),grid
```

该滤波器的频率响应以ω(rad/s)作为频率变量进行计算和显示，如图3.29所示，其与本章的表示方式保持一致。然而，在通信系统和数字信号处理领域，频率更多地采用f(Hz)表示。接下来的MATLAB程序以f作为频率变量。该程序计算并画出了该滤波器在频率范围$0\leqslant f<f_s/2$内1024个点处的频率响应。

```
fs=100;
num=0.1; den=[1 -0.9];
[H f]=freqz(num,den,1024,fs);
mH=abs(H); phH=angle(H)*180/pi;
subplot(2,1,1);plot(f,mH),grid
subplot(2,1,2);plot(f,phH),grid
```

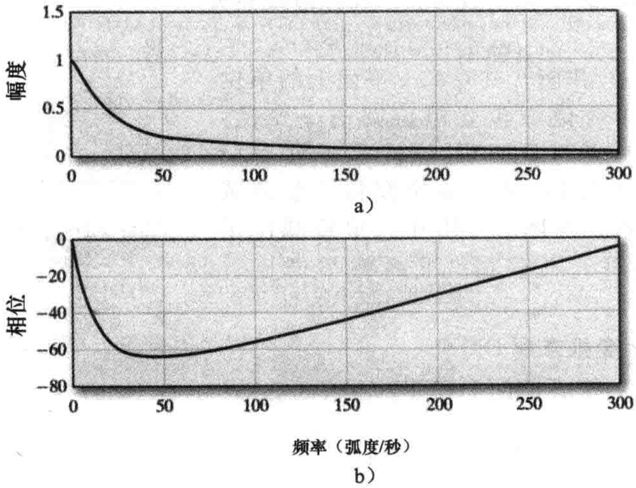


图3.29　例3-33的频率响应图 ◀

小结

本章研究了连续和离散时间信号的频域表示。傅里叶级数用于计算周期的连续时间信号的频谱；傅里叶变换用于计算非周期和周期的连续时间信号的频谱；离散时间傅里叶变换用于计算离散时间信号的连续频谱；离散傅里叶变换用于计算离散时间信号的离散频谱。快速傅里叶变换是计算离散傅里叶变换的快速计算方法。最后，拉普拉斯变换和z变换分别提供了连续时间系统与离散时间系统的复频域表示方法。

如果希望更深入地研究本章所提到的内容，建议查阅参考文献1～7。

习题

概念题

1. 为什么利用正交基函数(如傅里叶级数)表示信号是十分重要的？
2. 解释为什么在推导傅里叶级数系数时可以采用均方误差？还有哪些准则也可以是有效的？
3. 复指数函数在其派生的正交基函数中尤为重要的原因是什么？
4. $\sin(0)=0$，为什么$\text{sinc}(0)=1$？
5. 傅里叶变换中时间展缩性质体现了信号时间变化率与其频谱之间的什么对应关系？如何与系统的要求相联系？

6. 傅里叶级数与傅里叶变换之间的关系是什么？
7. 傅里叶变换与离散时间傅里叶变换之间的关系是什么？
8. 离散时间傅里叶变换(DTFT)与离散傅里叶变换(DFT)之间的关系是什么？
9. DFT 与 FFT 之间的关系是什么？
10. 解释 FFT 计算过程中补零的重要性。
11. 傅里叶变换与拉普拉斯变换之间的关系是什么？
12. 拉普拉斯变换与 z 变换之间的关系是什么？
13. DTFT 与 z 变换之间的关系是什么？
14. 解释 z 变换中需要知道收敛域的重要性。

计算题

3.1 节

3.1　周期信号三角形式傅里叶级数为

(i) $x_1(t)=3+5\cos t+6\sin\left(2t+\frac{\pi}{4}\right)$

(ii) $x_2(t)=-10+3\cos t+7\sin 4.5t$

(iii) $x_3(t)=5-10\cos(10^6 t)+4\cos(10^7 t)+2\sin(1.1\times10^7 t)$

(a) 计算每个信号的复指数形式傅里叶系数；

(b) 计算每个信号的组合三角形式傅里叶系数。

3.2　计算图 P3.2 所示周期信号的复指数形式傅里叶级数系数。

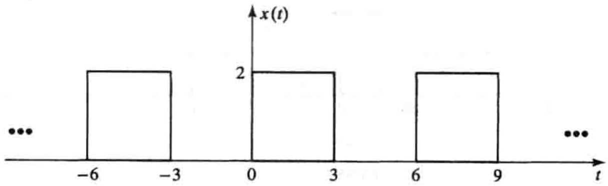


图 P3.2　周期矩形脉冲

3.3　(a) 求出所有使得 $\cos(m\omega_0 t)$和 $\cos(n\omega_0 t)$在周期 T_0 内相互正交的整数 m 与 n。

(b) 求出所有使得 $\cos(m\omega_0 t)$和 $\sin(n\omega_0 t)$在周期 T_0 内相互正交的整数 m 与 n。

(c) 求出所有使得 $\sin(m\omega_0 t)$和 $\cos(n\omega_0 t)$在周期 T_0 内相互正交的整数 m 与 n。

3.4　利用式(3.9)计算图 P3.4 所示信号的傅里叶系数。

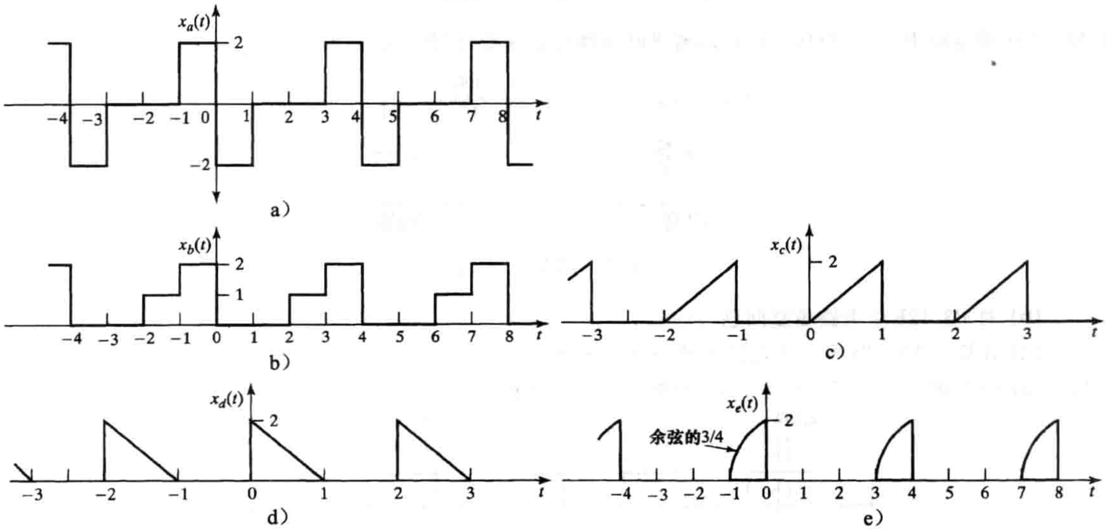


图 P3.4　周期信号

3.5 画出图 P3.2 所示信号的频谱，标出直流分量和前 4 个谐波分量。

3.6 (a) 画出图 P3.4a 所示信号频谱的幅度谱。标出直流分量和前 4 个谐波分量。

(b) 对图 P3.4b 所示信号重复问题(a)。

(c) 对图 P3.4(c)所示信号重复问题(a)。

(d) 对图 P3.4(d)所示信号重复问题(a)。

(e) 对图 P3.4(e)所示信号重复问题(a)。

3.7 利用 MATLAB 和傅里叶级数计算并画出幅度为 3 伏特的 10Hz 方波。利用不同 N 值的傅里叶级数，计算并画出至少一个完整周期的方波：(a)$N=3$，(b)$N=9$，(c)$N=33$.（注意，附录 D.1 节列出的 MATLAB 函数 fouriersq 可以有助于解答该问题。）

3.2 节

3.8 利用傅里叶变换的定义计算下列信号的变换：

(a) $x_a(t)=2[u(t-3)-u(t+3)]$ (b) $x_b(t)=2[u(t)-u(t-6)]$

(c) $x_c(t)=e^{-2t}[u(t)-u(t-6)]$ (d) $x_d(t)=t[u(t)-u(t-6)]$

(e) $x_e(t)=\cos(\pi t)[u(t+3)-u(t-3)]$

3.9 $x(t)=\cos(\omega_0 t)$的傅里叶变换为 $X(\omega)=\pi[\delta(\omega-\omega_0)+\delta(\omega+\omega_0)]$。利用以下性质，分别推导 $y(t)=\sin(\omega_0 t)$的傅里叶变换：

(a) 微分性质 (b) 时移性质

3.10 利用表 3.3 和表 3.4 求解习题 3.1 中信号的傅里叶变换。

3.11 计算图 P3.11a 和 b 所示信号的频谱，并进行对比。

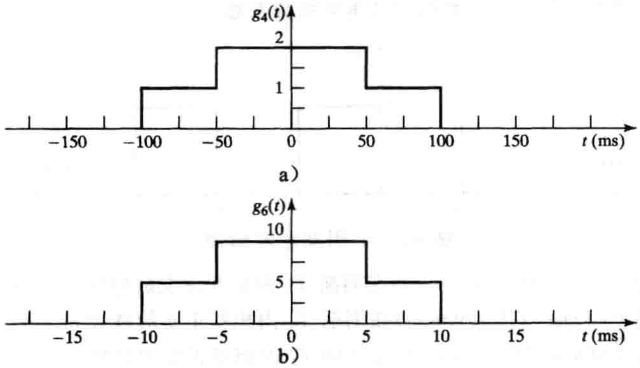


图 P3.11 习题 3.11 的信号

3.12 (a) 确定图 P3.12a 中所示电路的傅里叶变换传输函数(即频率响应函数)

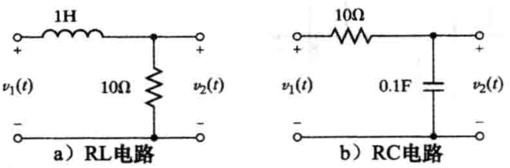


图 P3-12 习题 3.12 图

(b) 对 P3.12b 中电路重复问题(a)；

(c) 计算图 P3.12a 和 b 中电路的冲激响应函数。

3.13 (a) 计算图 P3.13 所示信号 $g_p(t)$的频谱，并画出其频谱图。

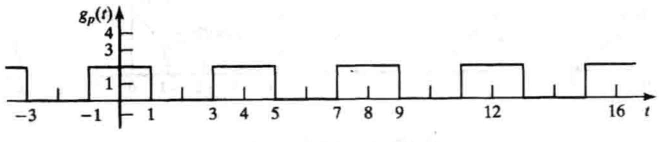


图 P3.13 周期矩形脉冲

(b) 如果 $g_p(t)$的周期加倍，频谱将如何改变？

(c) 如果 $g_p(t)$中每个脉冲的持续时间减半，频谱将如何改变？

3.14 计算信号 $x(t)=6\sin(100t)\cos(1000t)$的傅里叶变换，并画出其频谱图。

3.15 利用傅里叶变换计算卷积：

(a) $y_a(t)$中$=\text{sinc}(t) * \text{sinc}(t)$　　(b) $y_b(t)=\text{sinc}^2(t) * \text{sinc}(t)$

(c) $y_c(t)=\text{sinc}(t) * e^{j2t}\text{sinc}(t)$

3.16 利用时域卷积或者频域卷积定理计算以下信号的傅里叶变换，

(a) $3[u(t)-u(t-5)] * 2[u(t)-u(t-5)]$　　(b) $e^{-2t}\cos(100\pi t)u(t)$

(c) $\cos(100\pi t)\sin(1000\pi t)$

3.17 利用傅里叶变换的卷积性质，当 $v_1(t)$等于以下值时，求解图 P3.12 每个电路中的 $v_2(t)$：

(a) $v_1(t)=2\sin(50t)(\text{V})$　　(b) $v_1(t)=2\sin(100t)(\text{V})$

3.3 节

3.18 (a) 在 $N=8$ 时，利用 DTFT 求解具有下列脉冲响应的数字滤波器的频率响应。

$$h[n]=\begin{cases}1, & 0\leqslant n\leqslant N-1\\ 0, & 其他\end{cases}$$

(b) 画出幅频图。

3.19 (a) 利用 DTFT 计算以下信号的频谱：

(i) $x_a[n]=(0.5)^n u[n]$

(ii) $x_b[n]=(-0.5)^n u[n]$

(b) 利用 MATLAB 画出问题(a)中的幅频谱；　(c) 利用 MATLAB 画出问题(a)中的相频谱。

3.20 计算图 P3.20 所示各信号的 DTFT。

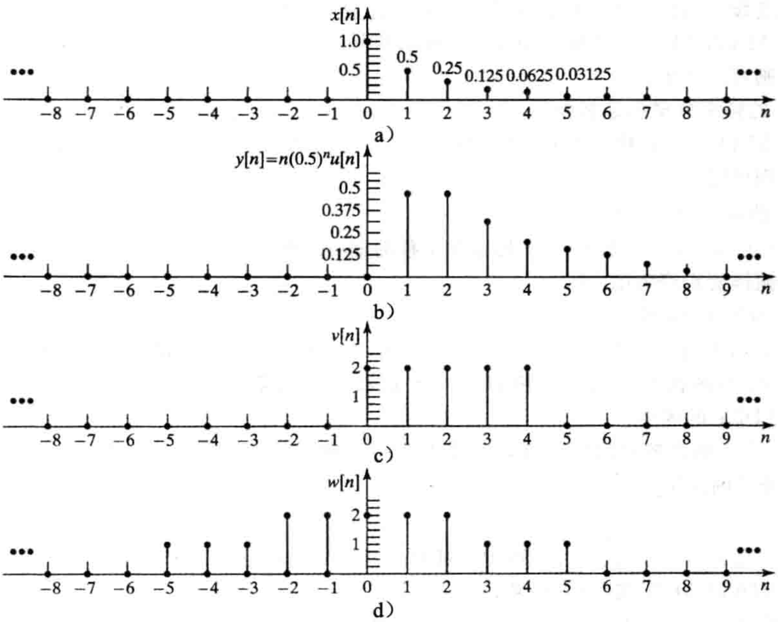


图 P3.20 离散时间信号

3.21 由下列差分方程描述的滤波器

$$y[n]=2x[n]+3x[n-1]+2x[n-2]$$

其中，$x[n]$为输入信号，$y[n]$为输出信号，

(a) 计算并画出此滤波器的脉冲响应 $h[n]$；　(b) 计算并画出 $H(\Omega)$。

3.22 给定离散时间信号的 DTFT 为 $X(\Omega)=\dfrac{1-e^{-j3\Omega}}{1-e^{-j\Omega}}$，输入到一个离散系统，其脉冲响应的 DTFT 为

$H(\Omega)=\dfrac{1}{1-0.5e^{-j\Omega}}$，

(a) 确定输出信号 $y[n]$的 DTFT； (b) 确定输出的离散时间信号 $y[n]$

3.23 计算周期 $N=5$ 的周期离散时间信号 $x[n]$的 $X(\Omega)$，其第一个周期的表示式为

$$x_0[n]=\begin{cases}1, & n=0,1,2\\ 0, & 其他\end{cases}$$

3.4 节

3.24 信号 $x(t)=u(t)-u(t-4)$如图 P3.24 所示。

(a) 计算从 $t=0$ 开始，以 $T_s=2\text{ms}$ 对其抽样，计算抽样后信号的 4 点 DFT，画出幅度谱和相位谱。

(b) 利用 MATLAB 计算并画出以 $T_s=1\text{ms}$ 抽样后信号的 8 点 DFT。

(c) 利用 MATLAB 计算并画出以 $T_s=0.5\text{ms}$ 抽样后信号的 16 点 DFT。

(d) 分析问题(a)、(b)和(c)的结果，讨论它们之间的关系。

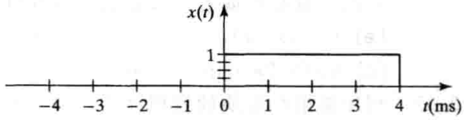


图 P3.24 矩形脉冲信号

3.25 (a) 计算图 P3.20a 所示信号的 8 点 DFT；

(b) 利用 MATLAB 程序验证(a)中结果。

3.26 (a) 计算图 P3.20b 所示信号的 8 点 DFT；

(b) 利用 MATLAB 程序验证(a)中结果。

3.27 (a) 计算图 P3.20c 所示信号的 8 点 DFT；

(b) 利用 MATLAB 程序验证(a)中结果。

3.28 对一个连续时间信号在 1 秒内进行 1024 点等间隔抽样，计算其 DFT。

(a) DFT 的相邻频率分量的间隔是多少弧度/秒？

(b) 在此连续时间信号中可以存在的最高频率分量应是多少？

3.29 (a) 利用 MATLAB 计算并画出图 3.20c 所示信号的 64 点 DFT。假设抽样周期为 $T=0.1\text{s}$，如果需要则可以补零。

(b) DFT 的频率分辨率是多少？

3.30 (a) 利用 MATLAB 程序计算并画出习题 3.24 中信号的 DFT，其中抽样周期采用 $T=0.1\text{s}$，如果需要则可以补零。

(b) DFT 的频率分辨率是多少？

3.31 对信号 $x(t)=5\cos(8\pi t)$从 $t=0$ 开始以 0.1 秒的抽样周期进行 8 点抽样。

(a) 计算抽样所得序列的 DFT；

(b) 利用 MATLAB 验证(a)中结果；

(c) 计算 $x(t)$的傅里叶变换，并将其与(a)、(b)中结果做对比，解释其中的差异。

3.32 将习题 3.31 中的离散时间序列乘以一个 8 点汉明窗后重复习题 3.31 过程。讨论此 DFT 与习题 3.31中 DFT 的区别。

3.33 将习题 3.31 中的离散时间序列乘以一个 8 点汉宁窗后重复习题 3.31 过程。讨论此 DFT 与习题 3.31 中 DFT 的区别。

3.5 节

3.34 (a) 利用 4 点 FFT 信号流图计算图 P3.34 所示序列的 DFT，

(b) 利用 MATLAB 证实(a)中结果。

3.35 (a) 利用式(3.50)计算图 P3.34 所示序列的 8 点 FFT；

(b) 利用 MATLAB 证实(a)中结果；

(c) 将此结果与习题 3.34 的结果对比。

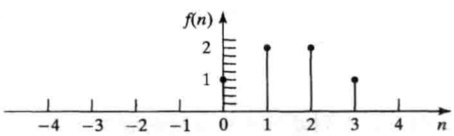


图 P3.34 离散时间信号

3.36 给出两个 4 点序列

$x[n]=[-2, -1, 0, 2]$和 $y[n]=[-1, -2, -1, -3]$

利用 DFT 求解。

(a) 循环卷积； (b) 线性卷积。

3.37 对于一个离散 LTI 系统，其输入信号为 $x[n]$，脉冲响应为 $h[n]$，输出信号为 $y[n]$。其中

$$x[n]=\begin{cases}e^{\frac{j2\pi n}{4}}, & n=0,1,2,3\\ 0, & 其他\end{cases}$$

$$h[n]=2\delta[n]+\delta[n-2],\quad n=0,1,2,3$$

(a) 计算 $N=4$ 时 $x[n]$的 DFT，$X[k]$；　(b) 计算 $N=4$ 时 $h[n]$的 DFT，$H[k]$；

(c) 计算 $x[n]\otimes h[n]$的 DFT，$Y[k]$。　(d) 计算 $y[n]$。

3.38 信号 $x(t)=3\cos(6\pi t)$从 $t=0$ 开始，以抽样周期为 0.1 秒抽样 128 个样本。

(a) 利用 MATLAB 计算抽样所得序列的 FFT；

(b) 计算 $x(t)$的傅里叶变换，并将其与问题(a)的结果对比，解释它们的差别。

3.39 将习题 3.38 中的离散时间序列与 128 点汉宁窗相乘后，重复习题 3.38 的问题，并讨论此 DFT 与习题 3.38 结果的差别。

3.40 将习题 3.38 中的离散时间序列与 128 点汉明窗相乘后，重复习题 3.38 的问题，并讨论此 DFT 与习题 3.38 结果的差别。

3.6 节

3.41 画出下列信号，并利用拉普拉斯变换的定义计算其拉普拉斯变换，

(a) $5u(t-2)$　(b) $3[u(t)-u(t-2)]$

(c) $3u(t-1)u(2-t)$　(d) $5u(a-t)u(t-b)$，其中 $a>b>0$

3.42 考虑由下列微分方程描述的连续 LTI 系统：

(i) $\frac{d^2y(t)}{dt^2}+5\frac{dy(t)}{dt}+4y(t)=2x(t)$　(ii) $\frac{d^2y(t)}{dt^2}+5\frac{dy(t)}{dt}+4y(t)=2\frac{dx(t)}{dt}+6x(t)$

(iii) $\frac{d^3y(t)}{dt^3}+3\frac{d^2y(t)}{dt^2}+4\frac{dy(t)}{dt}+2y(t)=6x(t)$　(iv) $\frac{d^3y(t)}{dt^3}-\frac{d^2y(t)}{dt^2}+2y(t)=4\frac{dx(t)}{dt}-8x(t)$

(a) 计算每个系统的传输函数；　(b) 计算每个系统的单位冲激响应 $h(t)$；

(c) 计算每个系统的单位阶跃响应 $s(t)$。

(d) 为了证实以上结果的正确性，验证(a)、(b)中的函数满足以下等式

$$h(t)=\frac{ds(t)}{dt}$$

从而将系统的冲激响应 $h(t)$与阶跃响应 $s(t)$联系起来。

(e) 利用 MATLAB 绘制(b)和(c)中的信号图形。

3.43 (a) 计算图 P3.43 所示系统的传输函数

(b) 求解系统的单位冲激响应；

(c) 计算系统的单位阶跃响应；

(d) 利用 MATLAB 绘制(b)和(c)中的信号图形。

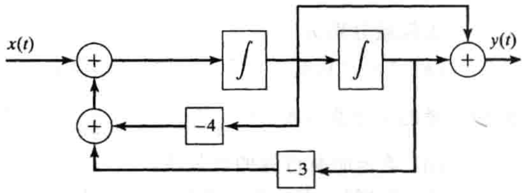


图 P3.43　二阶的连续时间反馈系统

3.44 计算并画出下列传输函数所描述系统的频率响应。

(a) $H_a(s)=\frac{20}{s(s+1)}$　(b) $H_b(s)=\frac{8s}{(s+1)^2}$

(c) $H_c(s)=\frac{s+2}{s^2}$　(d) $H_d(s)=\frac{2}{s^2(s+2)}$

3.45 利用拉普拉斯变换和 MATLAB 计算习题 3.42 中各个系统的频率响应。

3.46 利用拉普拉斯变换和 MATLAB 计算图 P3.43 中系统的频率响应。

3.7 节

3.47 给定离散时间信号

$$x[n]=a^nu[n]-b^nu[-n-1]$$

(a) 确定 a、b 的值以保证其 z 变换存在；

(b) 假设 z 变换存在，计算 z 变换及其收敛域。

3.48 计算下列信号的 z 变换及其收敛域：

(a) $0.9^nu[n]$　(b) $0.9^nu[n+1]$

(c) $0.9^nu[n-2]$　(d) $0.9^nu[-n-1]$

(e) $0.9^{-n}u[n+1]$　　(f) $0.9^{n}u[-n]$

3.49　计算下列每个离散时间序列的 z 变换及其对应的收敛域，

(a) $x_a[n]=0.5^n u[n]$　　(b) $x_b[n]=-0.5^n u[-n-1]$

(c) $x_c[n]=-0.5^n$　　(d) $x_d[n]=2\left(\frac{1}{2}\right)^n u[n]+3\left(\frac{1}{3}\right)^n u[n]$

3.50　系统传输函数的通用形式为⊖，

$$H(z)=\frac{b_0+b_1z^{-1}+b_2z^{-2}+\cdots+b_Mz^{-M}}{a_0+a_1z^{-1}+a_2z^{-2}+\cdots+a_Nz^{-N}}$$

证明当 $a_0\neq 0$ 时，系统是因果的。(考虑系统脉冲响应。)

3.51　(a) 给定离散时间信号

$$x[n]=\begin{cases}(0.5)^n, & -10\leqslant n\leqslant 20\\ 0, & 其他\end{cases}$$

计算其闭合形式的 z 变换；

(b) 确定(a)中 z 变换的收敛域；

(c) 对下列离散时间信号重复(a)、(b)；

$$x[n]=\begin{cases}(0.5)^n, & -10\leqslant n\leqslant 10\\ (0.25)^n, & n\geqslant 2\\ 0, & 其他\end{cases}$$

(d) 对下列离散时间信号重复(a)、(b)。

$$x[n]=\begin{cases}(0.5)^n & -10\leqslant n\leqslant 0\\ (0.25)^n, & 1\leqslant n\leqslant 10\\ 0, & 其他\end{cases}$$

3.52　(a) 确定图 P3.52 所示系统的 z 变换传输函数，

(b) 该系统是稳定的吗?

3.53　求解下列 z 变换对应的 z 反变换

$$X(z)=\frac{0.6z}{(z-1)(z-0.6)}$$

收敛域分别为：

(a) $|z|<0.6$　　(b) $|z|>1$　　(c) $0.6<|z|<1$

3.54　考虑 z 变换，$X(z)=\dfrac{2z}{z-1}+\dfrac{4z}{z-0.9}-\dfrac{z}{z-0.85}$

(a) 确定所有可能的收敛域；

(b) 根据(a)中的各个收敛域求解其对应的 z 反变换。

3.55　(a) 求解图 P3.35 所示系统的 z 变换传输函数

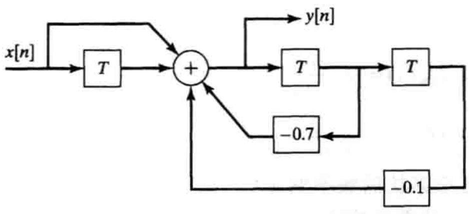


图 P3.52　一个带反馈的三阶离散时间系统

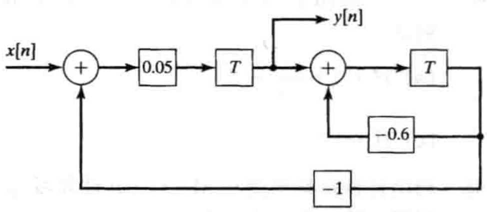


图 P3.55　一个带反馈的二阶离散时间系统

(b) 确定图 P3.55 所示系统的脉冲响应；

(c) 利用 MATLAB 画出此系统的单位阶跃响应；

(d) 计算这个系统所对应逆系统的**传输函数**，判断该逆系统是否满足以下条件：

(i) 是因果的吗?

(ii) 是稳定的吗?

⊖ 没有给出 Roc，也没有给出 M、N 的关系。——译者注

第4章

抽样与重建

前面章节利用传输函数定义数字滤波器，传输函数为两个含 z 的多项式之比，并对其在时域和频域进行了描述。本章将扩展上述定义，介绍更多与实现相关的概念。尽管某些系统仅用来处理数字信号，然而更常见的情形却是系统的输入和输出一般为模拟信号。图 4.1 给出了通用系统的框图结构。

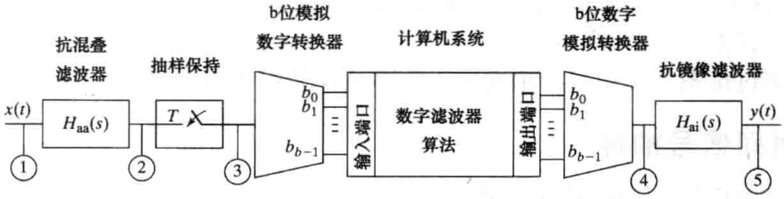


图 4.1　输入和输出端为模拟信号的数字滤波器

在系统最前端，模拟信号 $x(t)$ 通过低通抗混叠滤波器，之后抽样保持电路每 T 秒对信号抽样一次，抽样点应保持一定的时间，以允许模-数(A/D)转换器将其转换为数字信号，数字信号一般为正值的二进制数，范围一般为 $0\sim 2^{b-1}$ 之间，b 为 A/D 的位数。数字滤波器通常为计算机或微处理器中的软件程序，其接收的输入信号为 nT 时刻的原始信号 $x(nT)$。程序通过某种算法(通常为差分方程)计算 $y(nT)$ 的值，即 nT 时刻的输出。注意，由于在下一个周期中产生新的抽样值，需要在小于 T 秒的时间内完成 $y(nT)$ 的计算。$y(nT)$ 也是一个 b 位的二进制数值，通过数-模(D/A)转换器将其还原为模拟信号。通常需要通过寄存器保持 D/A 的输出值 $y(nT)$，直到下一次输出的到来。D/A 的输出通常是阶梯状的，如图 1-6 所示，将该阶梯状的输出通过另一低通模拟滤波器进行平滑，产生具有连续时间信号特征的信号，该滤波器称为抗镜像滤波器。图 4.2 给出了当信号通过数字滤波器系统各个阶段时，信号时域和频域变化的示意图。

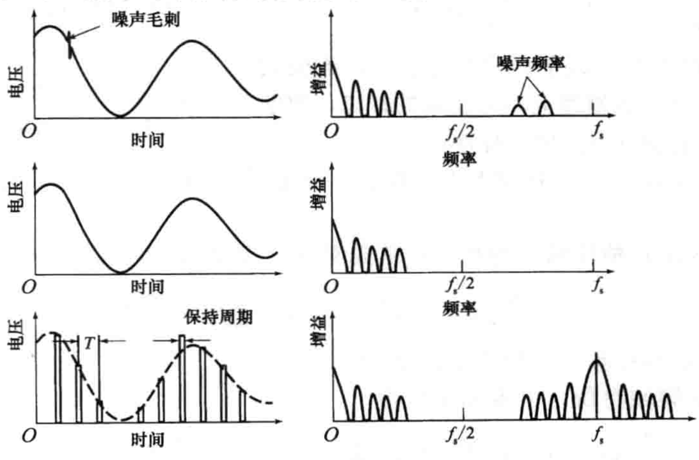


图 4.2　信号通过数字滤波器的时域和频域变化

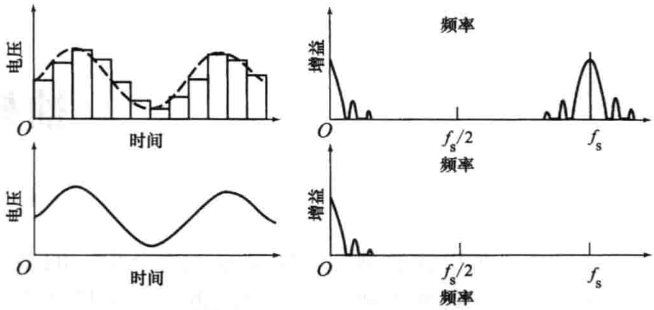


图 4.2 （续）

系统中的每一个阶段都会引入一些误差，如因有限的计算精度引起的差分方程计算误差，A/D 和 D/A 自身存在的转换误差，前端的抽样保持以及前端和后端的模拟滤波器也可能存在潜在的误差。

本章将详细讨论图 4.1 中各个单元的工作原理及其引入的误差。

4.1 连续时间信号抽样

对于连续时间信号 $x(t)$，抽样定义为特定时刻 $x(t)$的值组成的有序时间序列，抽样后产生的时间序列为 $x(t_1)$，$x(t_2)$，$x(t_3)$，…其中 t_i 为抽样发生的时刻，现仅考虑在某些固定时刻上的 $x(t)$。

举一个信号抽样的实例，信号抽样由硬件实现的 A/D 完成。A/D 是对连续时间电压信号抽样的电子器件，实际应用中通常用于信号的抽样，抽样后的信号可以由计算机进行处理。

在大多数情况下，以相等的时间增量抽样连续时间信号，抽样频率用 f_s 表示，抽样周期用 T 表示，抽样后的信号值用 $x(nT)$表示，其中 n 为整数。

4.1.1 冲激抽样

理想的冲激抽样过程如图 4.3 所示，其中，连续时间信号 $x(t)$与冲激串 $\sum\limits_{n=-\infty}^{\infty}\delta(t-nT)$ 相乘产生抽样后信号 $x_s(t)$，连续时间信号在 nT 时刻的信号值作为每个冲激信号的权重。

$$x_s(t)=x(t)\sum_{n=-\infty}^{+\infty}\delta(t-nT)=\sum_{n=-\infty}^{+\infty}x(nT)\delta(t-nT)$$

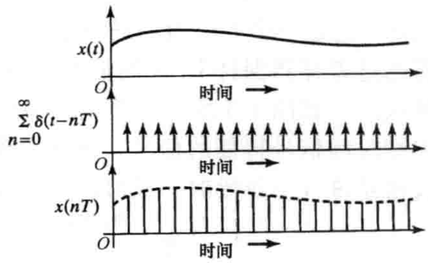


图 4.3 抽样信号表示为连续时间信号与冲激串的乘积

观察 $x_s(t)$，可以得出以下两个结论。一，该模型引入了冲激信号，因而是物理上不可实现的；二，数学意义上的抽样并没有得到所需的抽样序列 $x(nT)$，而只是加权的冲激串。尽管 $x_s(t)$是物理上不存在的，但使用冲激函数进行抽样操作在数学上是可行的。

为研究图 4.3 所示抽样操作的性质，首先对$x_s(t)$进行傅里叶变换，从例 3-6 可知

$$\delta_T(t)=\sum_{n=-\infty}^{\infty}\delta(t-nT)\xleftrightarrow{F}\omega_s\sum_{k=-\infty}^{\infty}\delta(\omega-k\omega_s) \tag{4.1}$$

其中，$\omega_s=2\pi/T$ 是抽样频率，单位是弧度/秒。

时域相乘对应频域卷积，由表 3.4 可得

$$X_s(\omega)=\frac{1}{2\pi}X(\omega)*\left[\omega_s\sum_{k=-\infty}^{\infty}\delta(\omega-k\omega_s)\right]=\frac{1}{T_s}\sum_{k=-\infty}^{\infty}X(\omega)*\delta(\omega-k\omega_s) \tag{4.2}$$

根据冲激函数的卷积性质

$$X(\omega) * \delta(\omega - k\omega_S) = X(\omega - k\omega_S)$$

式(4.2)中冲激调制信号的傅里叶变换可以写成

$$X_S(\omega) = \frac{1}{T}\sum_{k=-\infty}^{\infty} X(\omega - k\omega_S) \tag{4.3}$$

式(4.3)给出了抽样后信号相对于原始信号的频域特征。

考虑第一种情况，信号 $x(t)$的频谱是有限的，即对于 $\omega > \omega_B$ 时，$X(\omega)=0$。若 $\omega_B < \omega_S/2$，如图 4.4a 所示。从式(4.3)可以看出，对信号 $x(t)$进行抽样的结果为在 $k\omega_S$频率处对原始信号频谱的复制，$k=\pm1$，±2，±3，…在上述情况下，复制的频谱在频域内没有产生混叠，可使用截止频率为 ω_B 的理想低通滤波器恢复原信号。

在图 4.4b 中，信号的频谱仍是有限的，但最高频率 ω_B 大于抽样频率 ω_S 的一半，即 $\omega_B > \omega_S/2$，导致复制的频谱之间产生重叠。在重叠区域，频谱混叠在一起，造成信息丢失。截止频率为 ω_B 的理想低通滤波器的输出，将为原始频谱的一部分以及与其邻近频谱混叠的一部分，原始信息不能恢复。

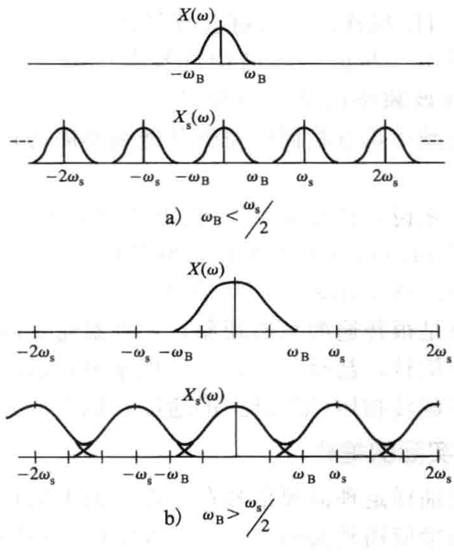


图 4.4 a) 连续时间信号的频率响应，信号的带宽 $|\omega| < \omega_B$，且 $\omega_B < \omega_S/2$。b)与 a)中的信号相同，但 $\omega_B > \omega_S/2$，频谱出现混叠，信息丢失

$\omega_S/2$ 称为奈奎斯特频率，抽样不失真的基本要求为，抽样频率满足 $\omega_S > 2\omega_M$，其中 ω_M 为待抽样信号的最高频率，该结论称为香农抽样定理(Shannon's sampling theorem)。

4.1.2 香农抽样定理

对于连续时间信号 $x(t)$，其最高频率为 f_MHz，可使用抽样间隔为 $T_M/2$ 秒($T_M=1/f_M$)的 $x(t)$的样点集唯一确定。

根据香农抽样定理，在 $x(t)$最高频率分量对应的周期中至少得到两个抽样值。在图 4.5 中有三个正弦信号，频率分别为 200Hz、800Hz 和 1200Hz，三个信号的抽样频率均为 1000Hz，我们发现在此抽样频率下，三个信号产生了相同的抽样值。若计算机从 A/D 中接收到这样的信号，则其无法判断输入信号的频率，因此，1200Hz 和 800Hz 信号对 200Hz 信号的识别产生了混叠。

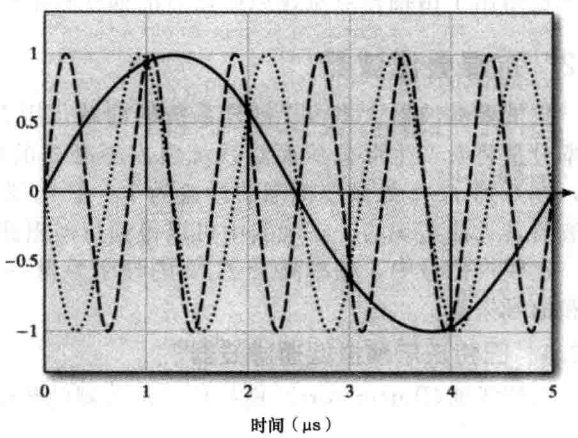


图 4.5 对频率为 200Hz、800Hz 和 1200Hz 的正弦信号，以 1000Hz 抽样，三个信号产生了相同的抽样值

以下列方式给出混叠较为正式的定义：

对于频率为 fHz 的正弦信号，若抽样频率为 f_SHz，存在无限多的其频率为 $|f+kf_S|$ 的正弦信号，这些信号将产生与该信号相同的抽样值，其中 k 是任意非零的整数。

例 4-1 混叠

对于 8kHz 的正弦信号，以抽样频率为 44.1kHz 进行抽样，找到与 8kHz 信号发生混叠的最低的 5 个正弦信号频率。

解： 想象一个理想化的频谱，其增益在 0～1 之间，假设信号频谱在 0Hz 处的增益为 1，在 $f_s/2$Hz 处增益为 0，对信号以 f_sHz 抽样，频谱将会被复制，如图 4.6 所示。在原始频谱中选择某一频率 f_1，在该频率的增益值处画一水平直线，与该频率具有相同增益的其他频率即为该频率的混叠频率，在图 4.5 中，f_1 的混叠频率为 f_2，f_3，…，f_7…

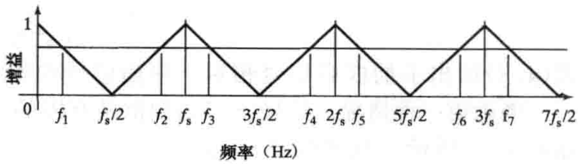


图 4.6 使用理想化的信号频谱确定混叠频率

一般来说，任意频率 f_1 的混叠频谱为 $kf_s \pm f_1$，其中 $k=1, 2, 3, \cdots$

使用 44.1kHz 频率抽样，8kHz 信号的前 5 个混叠信号频谱为 36.1kHz、52.1kHz、80.2kHz、96.2kHz 和 124.3kHz。◀

混叠是很普遍的自然现象，如观察轮子高速运转的马车，会发现车的轮子似乎以较低的速度在旋转，甚至后退，这种现象是相机的抽样速率较低而造成的。同时，如果使用闪光灯测量旋转物体(如飞轮)的速度，也会产生混叠。

4.1.3 实际的抽样

香农抽样定理需要信号在全部时间上的抽样值，包含过去和未来时间上的信号值，因此不能直接应用到实际中。而且在实际应用中，通常将高于感兴趣频率 4～10 倍的频率设为抽样频率 f_s。

频域受限的信号在时域不可能也是有限的，真实存在的信号在时域均是有限的，因而在频域是无限的，如图 4.4a 所示。然而，对于真实的物理信号，其高于某频率的幅度值将变得很小，足以忽略那些分量。在真实抽样中，要求频率高于 $f_s/2$ 的频谱幅度可以忽略。

注意傅里叶变换在确定抽样特征中的重要性，频域可清晰地显示信号抽样后的效果，同时也给出了由抽样点重建原始信号的抽样频率要求。

4.2 抗混叠滤波器

抗混叠滤波器大多位于抽样系统的前端，用于去除频率高于 $f_s/2$ 的频率分量，这些频率分量将会产生混叠从而给系统引入不必要的噪声。理想的抗混叠滤波器是低通滤波器，在 0 到 $f_s/2$ 处频率范围的增益为 1，在 $f_s/2$ 到无穷处频率范围的增益为 0。这样的滤波器是无法实现的，在现实中可以得到与理想滤波器近似的滤波器。

在模拟信号中，频率高于 $f_s/2$ 的频率分量一般来源于串音干扰、杂散磁场、电源干扰和信号失真。

4.2.1 巴特沃斯模拟低通滤波器[⊖]

巴特沃斯(Butterworth)模拟低通滤波器的传输函数为：

$$H_B(\omega) = \frac{1}{\sqrt{1+(\omega/\omega_c)^{2N}}}$$

其中，N 为滤波器的阶数，ω_c 是以弧度/秒表示的截止频率。

⊖ 附录 A 给出了模拟滤波器的详细介绍。

图 4.7 给出了不同阶数的巴特沃斯滤波器，在 1Hz($f_s/2$)处的增益为 0.5，由该图可知：第一，频率高于 $f_s/2$ 的分量被衰减，并没有去除。频率范围在 $f_s/2<f<f_s$ 的噪声将产生明显的混叠效应，滤波器阶数越低，混叠现象越明显。第二，数字滤波器的输入信号并不是 $x(t)$ 的抽样值，而是 $x(t)$ 通过抗混叠滤波器后的抽样值。对于频率稍小于 $f_s/2$ 的有用频率分量，通过抗混叠滤波器时会有比较明显的衰减。

由于巴特沃斯滤波器通带较为平坦，通常使用其作为抗混叠滤波器，但为获取较窄的过渡带，也可使用其他滤波器。所有模拟滤波器都会受到温度和老化的影响，同时很难在较低的价格上获取较高的精度。

目前广泛采用的解决抗混叠滤波器问题的方案有两种，第一种是使用开关电容滤波器，第二种是对输入信号进行过采样。

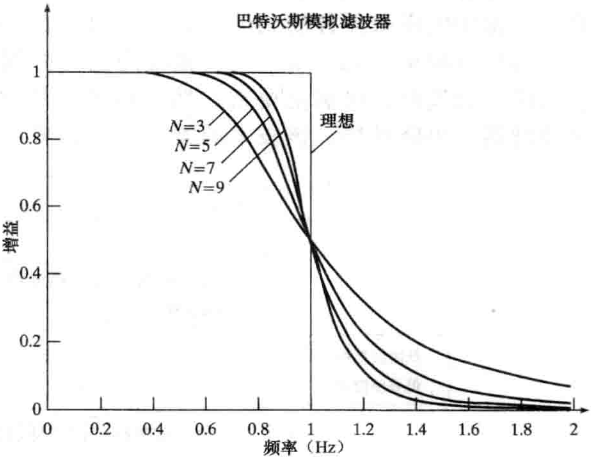


图 4.7　阶数为 3、5、7、9 的巴特沃斯模拟滤波器和理想低通滤波器的比较

4.2.2　开关电容滤波器

当构建集成电路时，很难得到滤波器中所需的精确电阻、电容和电感。将两个器件以精确的相对值构建比以精确的绝对值构建相对容易一些，例如，构建两个电容器，其中一个电容值是另外一个的 2/3 比较容易实现，但构建两个电容值分别为 3nF 和 2nF 的电容器相对较难一些。目前已有由开关电容组成的电路，该电路允许电容器以电阻器的方式工作，电路输出值依赖于电容器的相对值，而非绝对值。

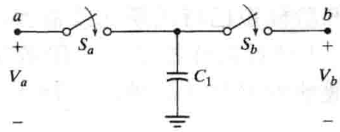


图 4.8　以电阻器方式工作的开关电容

在图 4.8 中，如果开关 S_a 闭合，开关 S_b 打开，电容器的充电量为 $q_a=C_1V_a$。如果开关 S_a 打开，开关 S_b 闭合，电容器的充电量为 $q_b=C_1V_b$。a、b 两点间的电压为 V_a-V_b，由 a 转移到 b 的电量为 $q_a-q_b=C_1(V_a-V_b)=\Delta q$，传输这些电量的时间为 Δt。由于 $\Delta q/\Delta t$ 是电流，可以写成 $i=\dfrac{\Delta q}{\Delta t}=\dfrac{C_1(V_a-V_b)}{\Delta t}$。由于电压和电流的比值为电阻值，因此有：

$$R_{eq}=\frac{\Delta t}{C_1}$$

如果该电路每 Δt 秒改变一次，且 $f=1/\Delta t$，可以得到

$$R_{eq}=\frac{1}{C_1 f}$$

将电容以电阻方式工作的概念使得某些集成电路的制作更容易且更统一，按照这种原理设计的滤波器称为开关电容滤波器，由于其精度高，价格低，常用来作为抗混叠滤波器。常用的开关电容滤波器为 2 阶到 8 阶，根据需要，也将几个滤波器进行级联。以 MAXIM 为例，MAX7420 为 5 阶的巴特沃斯低通滤波器，其通带截止频率用户可调，频率范围在1～45kHz之间[1]。该器件为普通的 8 脚封装，需要两个外部电容和一个外部晶体振荡器，用于确定通带截止频率。类似的器件还有 5 阶贝塞尔或椭圆低通滤波器，这些器件均可级联使用。

4.2.3　过抽样

过抽样也可用于解决抗混叠问题，其将问题的解决方式由硬件转移到软件。假设系统需要 8 阶巴特沃斯低通滤波器作为抗混叠滤波器，硬件上可以通过模拟器件实现，但价格较高，且常用的模拟器件受到老化、温度和难以获得所需精度等影响。可选择的解决方案为，提高抽样频率，如 4 倍，则过渡带的范围得到扩展，变为 $f_s/2 \sim 4(f_s/2)$，使用相对便宜的模拟滤波器即可满足要求。然后将过抽样信号通过 $f_s/2$ 处过渡带很窄的 8 阶低通数字滤波器，再降抽样。该技术的原理图，如图 4.9 所示。

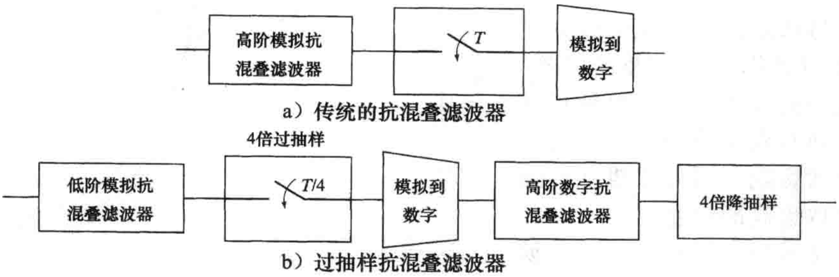


图 4.9　抗混叠原理

例 4-2　已知某数字信号处理器(DSP)系统的抽样频率 $f_s = 11.025$kHz，设计如下指标的抗混叠滤波器：

通带范围：0～3.85kHz

通带增益：最小值为 0.95，最大值为 1.0

阻带范围：6.615kHz 到无穷

阻带增益：最大值为 0.05

可使用 8 阶模拟巴特沃斯滤波器实现满足上述要求的滤波器，或者使用 3 阶模拟巴特沃斯滤波器，4 倍过抽样技术和 8 阶数字巴特沃斯滤波器实现(按照图 4.9b)的原理)。低通模拟抗混叠滤波器是 3 阶模拟巴特沃斯滤波器，如图 4.10 所示。

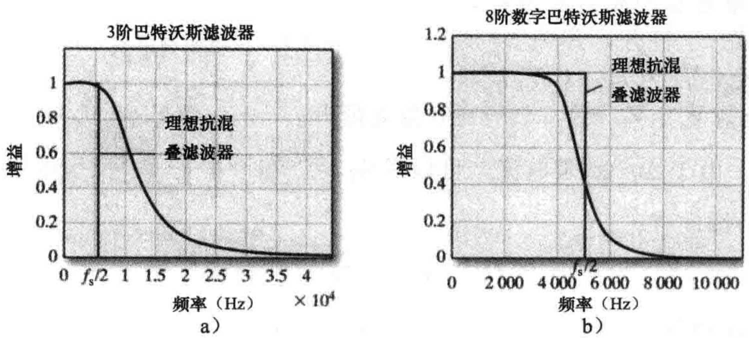


图 4.10　左图为 3 阶模拟抗混叠滤波器，之后需要 4 倍的过抽样器和 8 阶的数字巴特沃斯滤波器(右图)

图 4.11 为两个滤波器结合使用的结果，通带和阻带以子图显示，以验证该滤波器满足题目的要求。◀

过抽样系统的优点是可使用价格较低的器件实现模拟滤波器部分，同时当系统要求改变时，调整起来比较方便。

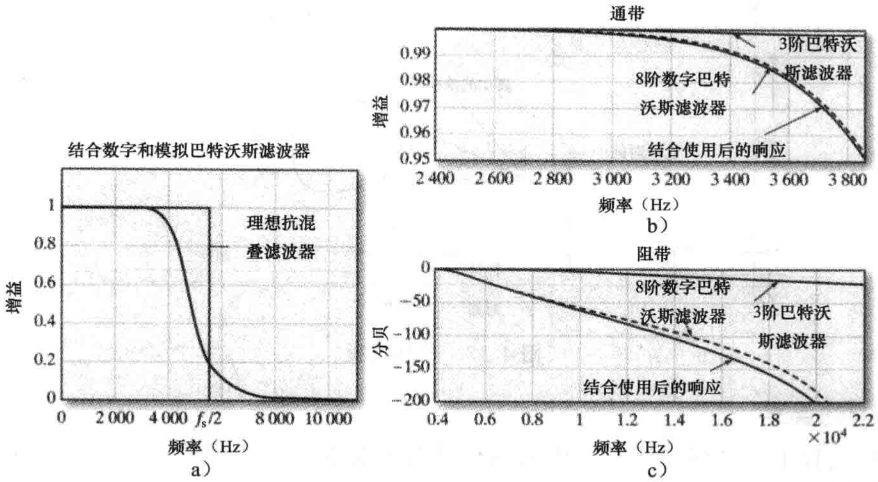


图 4.11　左图为 3 阶模拟滤波器和 8 阶数字滤波器结合后的幅度谱，右图为通带和阻带

对模拟抗混叠滤波器要求的降低并不是 A/D 中采用过抽样技术的唯一优点。由于过抽样过程中没有增加噪声的功率，即总的噪声功率没有改变，但其频率分布范围更广，噪声密度降低，因而可认为降低了量化噪声，K 倍过抽样将噪声密度降低了 K 倍。

更多关于过抽样的介绍可参考 7.9 节。

4.3　抽样过程

抽样器将一个时间连续的信号转换为时间离散的信号，在相同的抽样频率 $f_s=1/T$ 上，抽样过程基本是相同的，可以使用图 4.3 所示的脉冲序列表达抽样后的信号。

$$x(nT)=\sum_{n=-\infty}^{\infty}x(t)\delta(t-nT)=\sum_{n=-\infty}^{+\infty}x(nT)\delta(t-nT)$$

其中，$x(t)$为连续时间信号，$x(nT)$为抽样后信号，注意，抽样后得到的 $x(nT)$信号在时间上为离散的，但在幅度上仍是连续的，A/D 或者量化器可将信号变为幅度离散信号。

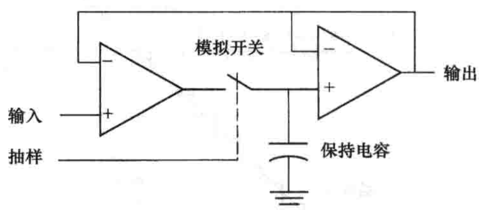


图 4.12　抽样保持电路框图

对于数字滤波器，抽样器和 A/D 是同时工作的。抽样器需要完成两个功能，即对信号进行抽样和保持抽样值直到 A/D 将其转换为离散幅度的数字信号。抽样保持电路的功能结构如图 4.12 所示，在该图中，输入信号为时间和幅度均连续的信号，如果模拟开关闭合，输出跟随输入。当模拟开关断开，电容器将在短时间内保持抽样值，此值出现在输出端。当模拟开关再次闭合，输出在极短的时间再次获取输入值，保持该值直到下一次抽样，图 4.13 给出了工作原理。在抽样过程中，保持电容起到了至关重要的作用，需仔细选择其容值和类型以满足特定的要求。

如果保持电容过大，信号获取时间变长，但被保持电压的幅值下降减小。相反，如果电容过小，信号获取时间变短，但被保持电压的幅值下降增大。若相比于信号改变的速度，A/D 的速度足够快，则不需要抽样保持电路，闪存转换器一般是按这种方式工作的。

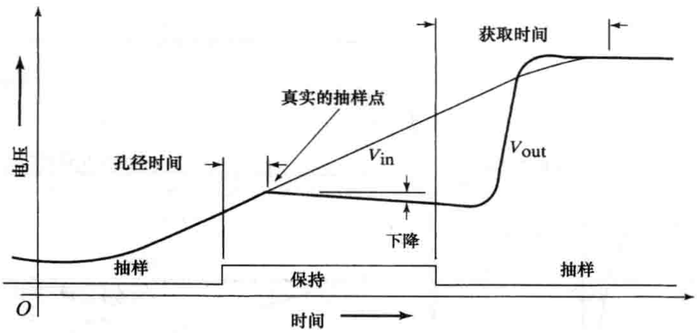


图 4.13　抽样过程

抽样过程中的误差

在抽样过程中，抽样器通常会引入两种类型的误差，一是抽样保持过程中保持电压的下降带来的误差，该误差为 1～2 位；二是抖动引起的误差，抖动是抽样周期的变化引起的，由热效应和电磁干扰造成的抖动误差符合高斯分布。该误差也可由常规的噪声源造成，如 60Hz 的电源设备。

这两种误差均在抗混叠滤波器后，造成了额外频率分量加到原信号中。

4.4　A/D 转换

A/D 也称量化器，其将时间离散信号变为时间和幅度均离散的信号，A/D 的典型参数包括转换位数和转换速率，如果信号被转换为 b 位，可用的量化等级为 2^b，A/D 的输出为最接近输入值的量化等级，图 4.14 给出了 4 位 A/D 转换器的量化结果。

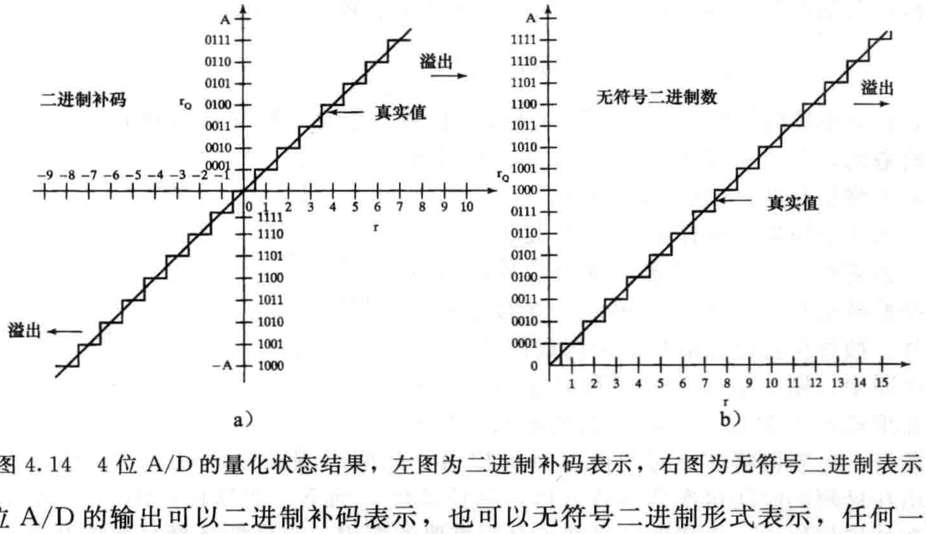


图 4.14　4 位 A/D 的量化状态结果，左图为二进制补码表示，右图为无符号二进制表示

b 位 A/D 的输出可以二进制补码表示，也可以无符号二进制形式表示，任何一种形式均可方便地转换为另一种形式。转换位数根据实际要求而定，对于灰度级的视频信号，4～6位 A/D 就足够了，而对于高保真的音频信号或者数字电压表需要 20 位，典型的音频文件使用 16 位 A/D。测量高转换精度对应的真实物理量是十分困难的，且价格较高，如 20 位 A/D 可以表达的物理量为 $1/2^{20}$，即百万分之一。

转换速率的差别也较大。数字电压表或微波炉温度的实时测量，转换器的转换时间为秒级。但对视频信号来说，转换时间为几十纳秒。对于数字滤波器来说，转换速率应至少与抽样周期相同，对于以 44.1kHz 抽样的音频信号，转换时间大约为 22.68μs。表 4.1给

出了一些常用的抽样频率及其对应的抽样周期。

表 4.1　常用的抽样频率及其对应的抽样周期

抽样频率/kHz	抽样周期/μs	应用	抽样频率/kHz	抽样周期/μs	应用
8	125	电话和无线话筒	88.2	11.34	专业音响
32	31.25	迷你 DV，如便携式摄像机	96	10.42	DVD 音频
44.1	22.68	音频 CD	192	5.21	DVD 音频
48	20.08	DVD 音响和数字录音带			

4.4.1　转换技术

下面将介绍三种常用的模拟数字转换技术以说明转换器的工作原理，分别为逐次逼近寄存器(Successive Approximation Register，SAR)技术，闪存转换器和 Σ-Δ 转换器。

4.4.2　逐次逼近转换器

逐次逼近转换器首先使用 D/A 对输入信号产生模拟预测，然后使用二分法对预测值进行修正，以找到正确的量化等级。对于 b 位转换器，一次完整的转换需要 b 次预测，这类似于使用乘法器进行除法运算。图 4.15 给出了 SAR 转换器的工作原理。

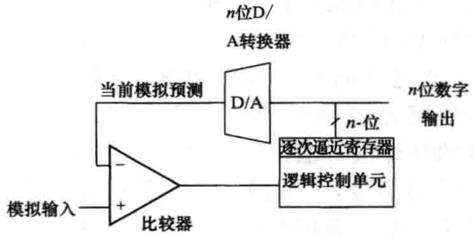


图 4.15　利用逐次逼近技术的 D/A 转换功能框图

当转换开始时，逻辑控制单元产生第一次逼近，逼近值为最大值的一半，D/A 将该逼近值转换为模拟信号，并将其与输入信号比较。比较器的输出告诉逻辑控制单元当前逼近值是高于输入还是低于输入，如果估计值过低，则下一次预测为最大值的 3/4，如果过高，下一次预测为最大值的 1/4，重复该过程直至 n 位。

这种类型的 D/A 速度较快，10～14 位的转换可在几微秒内完成，但其存在一个不足，由于每次都最先获得较高位的值，因此对于以较低位数表示的模拟值，需占用几乎完整的转换时间。

4.4.3　闪存转换器

闪存转换器使用并行转换技术，但其转换位数一般小于 10。图 4.16 给出了闪存转换器的功能框图。

对于图 4.16 中的闪存转换器，连接到参考电压上的电阻阶梯用于建立量化等级，对于 b 位转换器需要 2^b+1 个电阻，将输入的模拟信号同时与所有的量化等级进行比较，输入小于量化等级的比较器以一种方式触发，输入高于量化等级的比较器以另外一种方式触发，逻辑电路为严格的组合逻辑，并将 2^b 位的输入转换为 b 位的数值。

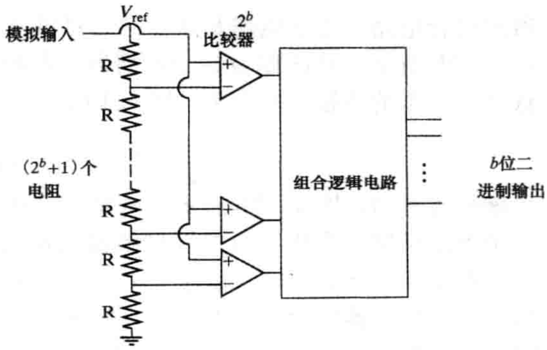


图 4.16　使用闪存转换技术的 A/D 功能框图

闪存转换器的缺点是使用了大量的电阻和比较器，转换位数每增加一位，电阻器和比较器的数量变为原来的 2 倍。优点是该转换器没有串行逻辑，因此转换时间很短。该转换器主要应用在转换速度要求高但精度要

求不是很高的场合，视频是其常见的应用场合。

由于闪存转换器的速度足够快，因此在大多数场合下，不需要抽样保持电路。

AD9066是Analog Devices[2]公司生产的双6位闪存转换器，单极供电电压为5V，每秒可完成6千万次抽样(million samples per second，MSPS)，对应的转换时间为16.67ns。

4.4.4 Σ-Δ转换

Σ-Δ调制是一种使用过抽样构建A/D或D/A的技术，其思想来源于1962年提出的Σ-Δ调制[3]。该技术以较高的速率和较低的精度(一般仅为1位)对模拟信号抽样。其对低分辨率信号进行时间上的平均，以提高降低的抽样频率下的精度。实际应用中，Σ-Δ A/D和D/A的转换技术适用于音频范围，精度为16～20位，在很多商业音频系统中有着广泛的应用，如CD播放器。

Σ-Δ转换器中的'Δ'指差分模块，其输出真实的输入值和当前输入值估计的差值。此差值信号可认为是经过脉冲编码调制(Pulse Code Modulation，PCM)后的输入信号，产生差值信号的电路称为Δ调制器，如图4.17所示。

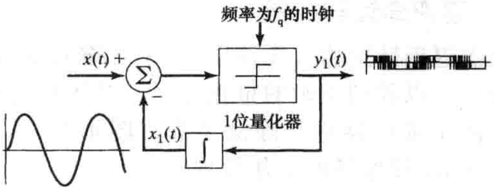


图4.17 Δ调制器，输入信号$x(t)$是正弦信号，估计值$x_1(t)$和调制输出$y_1(t)$如图所示，信号f_q为量化器的时钟

在图4.17中，输入到Δ调制器的信号为正弦波，1位量化器根据接收到的差值信号的正负，输出+1或−1。量化器的输出反馈到积分器，积分器将差值信号相加，产生输入信号的估计值，输出结果$y_1(t)$为Δ调制后的信号，'时钟f_q'决定量化的速率。积分器接收到的信号或者为+1，或者为−1，其按时钟频率改变速率。

若将Δ调制器的输出信号通过信道传输，则可使用图4.18所示的解调器恢复原始信号，在该电路中，使用积分器恢复被估计信号，低通滤波器阻止Δ调制器引入的高频分量，抽取器用于降低抽样频率，输出结果为原始信号的离散抽样形式。由于Δ调制器的输入为模拟信号，Σ解调器的输出为数字信号，若将Δ调制器和Σ解调器结合使用，可构成一个输入为模拟信号输出为数字信号的A/D转换器。

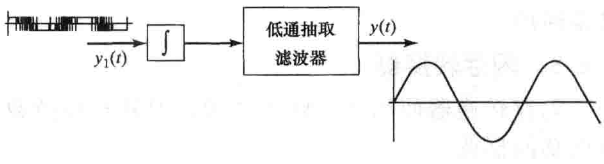


图4.18 解调电路将Δ调制后的信号转换为原始信号

将Δ调制器和Σ解调器组合一起，可以进行某些简化，如图4.19所示。

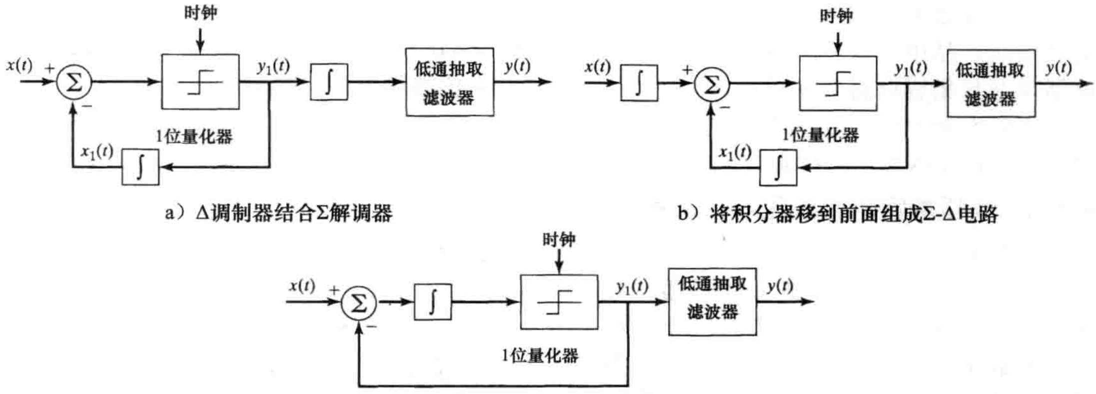
图4.19a为Δ调制器和Σ解调器的组合电路，由于电路是线性的，可以在不改变电路传输函数的前提下移动积分器，如图4.19b所示，该图称为Σ-Δ转换器。若积分器的传输函数为I，量化器的传输函数为Q，低通滤波器的传输函数为L，图4.19所示的电路的传输函数为：

$$H = I\frac{Q}{1+QI}L = \frac{QI}{1+QI}L$$

由该公式可知，图4.19b所示的电路可简化为图4.19c。简化后的图仅使用一个积分器和一个单位反馈，为传统的Σ-Δ转换器原理图。

在图4.19c中，输入信号$x(t)$、加法器和积分器均为模拟器件，为得到数字量输出，低通抽取滤波器应为数字器件。为实现量化器中的时钟，可在反馈通道中增加一个单位延时器，并将延时时间设置为时钟周期，量化器变为模拟比较器，改进后的结构如图4.20所示。'1位数字模拟转换器'表示的模块包含一个以时钟频率驱动和合适电平平移的单位

延时寄存器，电平平移的目的是将输出结果调整到输入范围的最高有效位，因此 D/A 的输出为输入信号的 1 位近似值。



a）Δ调制器结合Σ解调器　　b）将积分器移到前面组成Σ-Δ电路

c）去除一个积分器以简化图b

图　4.19

Σ-Δ 的时钟频率是很高的，因而在很多应用中，模拟输入端并不需要抽样保持电路。1 位 D/A 的时钟频率 f_Σ 一般高于输入信号抽样频率 f_s 的许多倍。

在图 4.19c 中，可将 1 位量化器看成系统中引入噪声的单元，如图 4.21 所示。

从该图可以得到两个传输函数：

如果噪声为 0，则 $H_1(s)=\frac{y_1}{x}=\frac{1}{s+1}$。

如果输入信号 $x(t)$ 为 0，则 $H_2(s)=\frac{y}{N}=\frac{s}{s+1}$，其中，$N$ 是噪声信号。

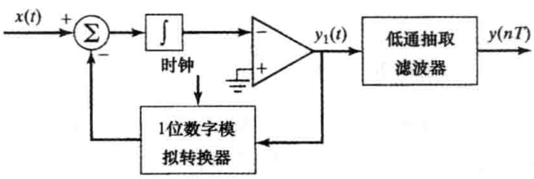


图 4.20　反馈通道中带有 1 位 D/A 转换器的 Σ-ΔA/D

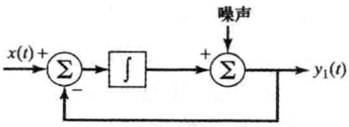


图 4.21　将图 4.19c 中的 1 位量化器替换为加性噪声源

从 x 至 y_1 之间的传输函数为低通滤波器，从 N 至与 y_1 之间的传输函数为高通滤波器。低通滤波器允许信号通过，高通滤波器允许噪声通过，通过附加的低通抽取滤波器可将信号和噪声分离，该过程称为*噪声整形*。噪声整形使用低通滤波器移除高频分量，同时保持信号的完整性，这也正是 Σ-Δ 转换器在较高抽样频率上能保持较高精度的原因。

在 A/D 输出端的低通抽取滤波器有两个功能，一是去除混入输入信号中的高频分量，Σ-Δ 环路高抽样率以及环路的输出为脉冲信号而非平滑的正弦信号，因而引入了高频分量。二是将 Σ-Δ 环路的抽样率降为所需的抽样率，该过程称为抽取。在大多数的实际应用中，常将这两个功能集成到一个电路(第 8 章将介绍抽取滤波器)。

注意，如果正弦信号的频率 f 大于 $f_s/2$，在降抽样后的信号中将产生混叠。为了对频率分量高于 $f_s/2$ 的信号进行抽取，需在抽取前利用低通滤波器减小信号的带宽，这也是 Σ-Δ 转换器采用的方式。由于 Σ-Δ 环路使用了较高频率的时钟，引入了高频分量，因而在抽取前使用滑动平均滤波器以去除高频分量。

Σ-Δ 环路转换的优势为在于其使用的抗混叠滤波器非常简单，仅采用一阶阻容(RC)电路即可实现该滤波器。通常来说，如果信号中高于 $f_\Sigma/2$ 的频率分量被滤除，将不会产生混叠。满足该要求的滤波器在 0Hz 到 $f_s/2$ 处的增益为 1，在 $f_\Sigma/2$ 到无穷处的增益为 0，

由于 $f_\Sigma \gg f_s$，因而该抗混叠滤波器的过渡带($f_s/2$ 到 $f_\Sigma/2$)可以相当大，对于 $f_s/2$ 到 $f_\Sigma/2$ 的频率，在通过低通抽取滤波器的过程中被滤除。

采用 Σ-Δ 调制技术的模-数转换器称为 Σ-Δ 转换器，由于符号函数每次转换 1 位，在某些商业产品中，也将其称为"1 位"转换器。National Semiconductor 公司是制作 Σ-Δ 转换器单芯片集成电路的主要厂商之一，48kHz 的 18 比特精度的转换器很常见。

实际中常用的 Σ-Δ A/D 转换器元件包括 National Semiconductor 公司的 ADC12EU050，其为八进制转换器，串行输出，转换位数为 12 位，速度为 40MSPS，供电电压为 1.2V。

4.4.5 模拟数字转换过程中的误差

模拟数字转换过程会引入多种误差，最常见的误差为量化误差，该误差由使用有限的位数表示抽样后信号的结果而引起。对于 A/D，量化误差为原始模拟信号与量化后的数字信号的差值，图 4.22 所示为量化后的正弦信号及其产生的误差示意图。

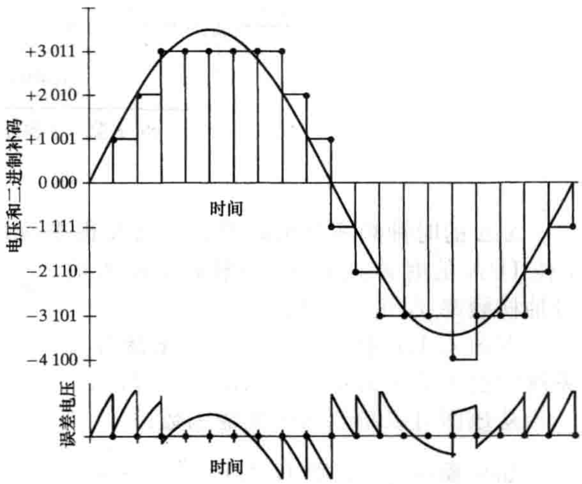


图 4.22 顶部的图为量化后的正弦信号，底部的图为量化误差

量化信噪比(Quantized Signal-to-Noise Ratio，QSNR)用于描述量化误差，其定义为

$$\text{QSNR} = 10\lg\frac{P_s}{P_n}\ \text{分贝}$$

在上式中，P_s 是信号功率，P_n 为噪声功率。对于幅度为 A 的正弦信号，平均功率为 $A^2/2$。为了确定噪声功率 P_n，需对噪声信号进行合理假设。对于 b 位 A/D 转换器，假设相对于信号频率，抽样频率足够高，则最大的量化误差将不会超过 $\Delta/2$，其中，$\Delta=2A/2^b$。由于正弦信号是对称的，且抽样间隔很小，将误差表示为在 $-\Delta/2\sim+\Delta/2$ 范围的直线，如图 4.23 所示。

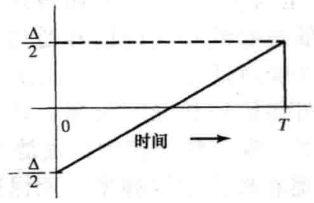


图 4.23 将误差信号假设为在一个采样周期上从 $-\Delta/2\sim\Delta/2$ 之间的直线

由于误差信号通常为电压信号，其平均功率 P_n 为误差信号的平方值在一个抽样周期上的平均，表示为：

$$P_n = \frac{1}{T}\int_0^T(\Delta t/T-\Delta/2)^2\,dt$$

积分结果为

$$P_n = \frac{\Delta^2}{12}$$

用 $2A/2^b$ 替换 Δ，得到

$$\frac{P_s}{P_n} = \frac{12(A^2/2)}{4A^2/2^{2b}} = \frac{3}{2}\cdot 2^{2b}$$

将量化误差表示为分贝：

$$\text{QSNR} = 10\lg(P_s/P_n) \approx 6.02b + 1.76 \tag{4.4}$$

该结果指出 A/D 中每增加一位，信噪比将提高 6db。在(4.4)式的计算中，我们假设输入信号为正弦信号，如果输入不是正弦信号，则 QSNR 为

$$\text{QSNR} \approx 6.02b + 10.81 - 20\lg(x_{\max}/P_s)$$

习题 4.30 给出了该结果的推导过程。

例 4-3 **对频率为 1kHz 的正弦信号进行抽样，抽样频率 $f_s=22050$Hz，使用 10 位 A/D 转换器，由于平均功率与信号的平方成正比，使用如下公式计算信号与噪声的平均功率**

$$P_{avg}=\frac{1}{N}\sum_{n=0}^{N-1}x^2(n)$$

使用 MATLAB 经功率计算得到 QSRN，并将结果与式(4.4)的计算进行比较。

解：如下 MATLAB 的 m 文件创建一个周期上的 1kHz 的正弦波，其抽样频率 $f_s=22050$kHz，量化信号的产生如下：将信号与 2^{b-1} 相乘，将相乘后的结果转换为其最接近的整数，再除以 2^{b-1}，即得到量化后信号。原始信号用 x 表示，量化后信号用 xq 表示，噪声信号为 $x-xq$。

```
%QSNR.m
b = 10; % 10 bit A/D
fs = 22050; % Sample rate
T = 1/fs; % Sample period
f = 1000; % signal frequency
N = fs/f; %Number of samples in one period
% x is the signal - a 1 kHz sin wave
% xq is the quantized signal - quantized to b bits
% Ps is the signal power. Pn is the noise power.
i = 1:N;
x = sin(2*pi*f*(i-1)*T);
xq = round(x.*2^(b-1)); % truncate by rounding
xq = xq/(2^(b-1));
Ps = sum(x.^2);
Pn = sum((x - xq).^2);
Ps = Ps/N;
Pn = Pn/N;
db = 10*log10(Ps/Pn);
fprintf('The QSNR is %f \n', db);
```

MATLAB 的输出结果 QSNR 为 62.27，如果使用式(4.4)计算，结果为 QSNR≈$6.02\times10+1.76=61.96$。◀

模拟数字转换的第二个重要误差为尺度误差，其经常被忽略。尺度化问题为：A/D 接收的输入信号范围需要在一个固定的电压范围，将其转换为 b 位。如果输入的信号没有占据满量程，则转换结果仅使用较少的位数。假设 A/D 接收信号的电压范围为 0～10V，转换位数为 10 位，则其能表达的最小值 Δ 为：

$$\Delta=10\text{V}/2^{10}=0.009766\text{V}$$

允许的偏移为半位，因而误差为 ±0.009766/2＝±0.004833，或 ±0.4833%。假设输入的范围不是 0～10V，而是该范围的一半，则误差变为原来的 2 倍，即 ±0.9766%。

A/D 转换的转换位数决定了转换器的分辨率，12 位转换器的分辨率为 2^{12} 分之一，但是分辨率并不能表示转换的精度，A/D 转换的精度表示为实际转换结果与其对应的理论上限值之间的差距。

A/D 中的误差分为静态误差和动态误差，静态误差是与频率无关的误差，而动态误差会随着频率的改变而改变，静态误差可以表示为 A/D 转换器输入电压的函数。

在 A/D 中，有 4 种主要的静态误差：(1)偏移误差，(2)增益误差，(3)微分非线性误差，(4)积分非线性误差。

- *偏移误差*——偏移误差独立于输入电压，可以通过在输入端输入0值，检测输出值进行测量。通常可以使用最低有效位(LSB)电压或满量程输出电压的百分比表示偏移误差。许多A/D均有参考电压连接端以允许用户调节输出电压。偏移误差是参考电压的函数，常见的偏移误差等于1/2 LSB。
- *增益误差*——增益误差与输入电压值成正比，同时也可能会随着抽样频率的变化而不同。其与偏移误差相同，增益误差也以满量程输出电压的百分比，或LSB电压表示。
- *微分非线性误差*——如果A/D的输入为斜坡信号，则输出应为包含所有可能结果的平滑阶梯，阶梯的过渡为1 LSB，微分非线性表示A/D完成上述功能的能力。有时某些量化阶梯可能丢失，或在阶梯过渡的地方出现，微分非线性误差给出了最坏的情况。微分非线性误差以LSB表示，如果A/D的微分非线性误差<±1 LSB，将不存在漏码，即当输入信号由最小值逐渐增加到最大值时，输出端将出现所有的二进制编码。
- *积分非线性误差*——该误差表示，在所有可能的输出中，真实输出与理想输出的最大方差，同微分非线性误差类似，积分非线性误差同样以LSB表示。

模拟数字转换中的动态误差是频率的函数，最重要的动态误差一般以QSNR表示，其为输入信号频率的函数。为计算该比值，信号为基频信号归一化后的平均功率，噪声是除基频、谐波和直流电源外的信号中所有噪声的归一化平均功率之和。最佳信噪比的理论值可由式(4.4)计算，QSNR≈6.02b+1.76。如果噪声信号包含所有谐波，则*信噪比*(Signal-to-Noise Ratio，SNR)变为*信噪比和失真*(Signal-to-Noise Ratio and Distortion，SINAD)，SNR和SINAD均可通过在A/D中输入精确的正弦信号，检测其输出信号的频谱得到。对于SINAD，输入信号应尽可能接近满量程，因为其表示信号失真时的最坏情况。通过准确地计算信号功率、噪声和谐波功率谱，可以得到SNR和SINAD。

一些制造商会提供A/D的有效位数(Effective Number Of Bit，ENOB)指标，其为SINAD的另一种表示形式，二者可以相互转换，根据式(4.4)，用SINAD的测量值替换QNSR，得到ENOB，如下

$$\text{ENOB}=\frac{\text{SINAD}-1.76\text{db}}{6.02}$$

MAXIM公司的MAX1425A/D转换器为该类型转换器，其技术指标如下：转换位数为10位，转换速率为20MSPS，10MHz下的典型SINAD为59db，ENOB为9.5。

例4-4 A/D转换器的前端包含传感器、多路复用器、单位增益缓冲放大器、抽样保持电路和A/D，假设复用器的电压误差为0.003%，放大器的电压误差为0.01%，抽样保持电路的误差为0.01%，如要求整个系统的误差小于50db，则该A/D的ENOB和SINAD应是多少？

解： 在"误差预算"为50dB的要求下，对A/D端输出的电压，需要满足

$$20\lg(v_0/v_i)<-50$$

求解上式，得到$v_0/v_i<0.003162$，其中，v_i表示来自A/D前端电路的噪声。

对于最坏的情况，单位幅值的信号通过复用器，放大器和抽样保持电路后的增幅值变为(1.00003 * 1.0001 * 1.0001)=1.000023，如果该信号通过A/D，则要求1.000023 * A/D的噪声增益<1.003162，因此A/D的噪声增益需小于1.00293，即1/2 LSB需要表示小于0.293%的信号，即

$$2\left(\frac{1}{2^{\text{ENOB}}}\right)<0.00293$$

$$2^{ENOB}>682$$

求解上式得，ENOB=10，代入式(4.3)，有 SINAD=61.96dB。

另一种方法为计算均方误差，该方法在现实中使用更多，但标准相对较低。对于这种方法，有

$$\sqrt{(0.00003)^2+2\cdot(0.0001)^2+(\text{A/D 误差})^2}<0.003162$$

求解上式得，A/D 误差<0.003145。

$$2\left(\frac{1}{2^{ENOB}}\right)<0.003145$$

$$2^{ENOB}>636$$

◀

4.4.6 抖动

抖动的概念并不直观，其主要思想是指当在信号中增加少量的随机噪声时，可以得到更好的信号量化结果。在对信号进行量化时，低于半位表示数值的所有值均被量化为较低的量化等级，假设系统为 3 位，表示的电压范围在 0～7.5V，每位表示的电压值为 $8/2^3=1.0$V，小于该值一半的电压($0\leqslant x<0.5$)均表示为 0，001 表示的电压范围为 0.5～1.5。3 位系统表示的所有数值和其对应的电压范围如表 4.2 所示。

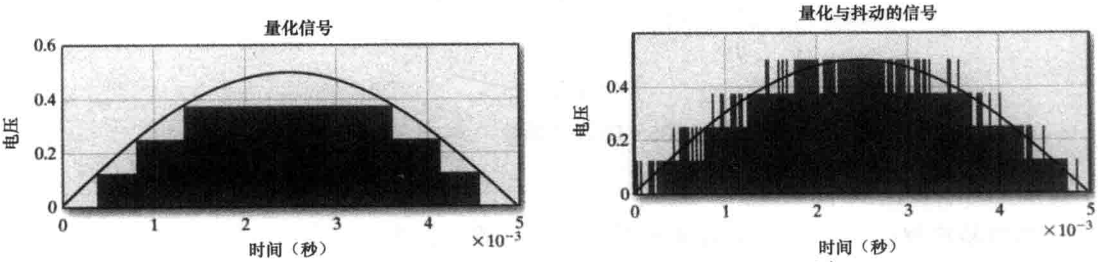
表 4.2　以 3 位表示电压值在 0～7.5V 的电压范围

二进制数	电压范围	二进制数	电压范围
000	$0\leqslant x<0.5$	100	$3.5\leqslant x<4.5$
001	$0.5\leqslant x<1.5$	101	$4.5\leqslant x<5.5$
010	$1.5\leqslant x<2.5$	110	$5.5\leqslant x<6.5$
011	$2.5\leqslant x<3.5$	111	$6.5\leqslant x<7.5$

在该系统中，任何小于 0.5V 的连续信号均表示为 000，但这些信号并不一定为零。若在该信号中加入少量噪声，在某些时间内，噪声可以增加信号的值，使其超过相邻的下一个更高的阈值，这样可以得到更加正确的表示。噪声必须是均匀分布的随机噪声，其值不能超过半位表示的电压值。增加的噪声称为抖动值，增加噪声的过程称为抖动。例 4-5 给出了抖动的工作原理，抖动的数学表示在 9.6 节将有详细介绍。

例 4-5 利用 MATLAB 产生半个周期的正弦波，正弦波的频率为 0.1kHz，抽样频率为 44.1kHz，使用 4 位量化该正弦信号，画出量化结果。在原始信号中加入抖动电压，重新进行 4 位量化，画出增加抖动后的量化结果，并将其与未抖动的量化结果比较。

解： 如下 MATLAB 的 *m* 文件产生了 0.1kHz 的正弦波，将其量化为 4 位。在该信号中增加抖动信号，再次进行 4 位量化，比较结果如图 4.24 所示。



a)　　b)

图 4.24　该图所示为量化为 4 位的 1kHz 正弦信号，在图 b 中加入了抖动

```
fs = 44100;T = 1/fs;
f0 = 100;
Bits = 4;
% Create a half cycles of a sine wave at f0.
i = 1:220;
u = 0.5*sin(2*pi*f0*T*i);
%uquant is the quantized version of u.
uquant = u*2^(Bits-1);
uquant = fix(uquant);
uquant = uquant/2^(Bits-1);
%
figure(1);clf;
subplot(2,1,1)
Time = linspace(0,219*T,220);
stem(Time, uquant, 'MarkerEdgeColor', 'none')
axis([0 .005 0 .6])
%
% rand produces a flat distribution between 0 and 1.
dither = rand(1,length(u))/2^(Bits-1);
%ditherU is the original signal plus the dither.
ditherU = u + dither;
%dquant is the quantized dithered signal
dquant =ditherU*2^(Bits-1);
dquant = fix(dquant);
dquant = dquant/2^(Bits-1);
subplot(2,1,2)
stem(Time,dquant, 'MarkerEdgeColor', 'none')
axis([0 .005 0 .6])
```

4.5 D/A 转换

在得到数字系统的输出后，经常需将其还原为连续时间信号，该过程的第一步为数字模拟转换(D/A)。D/A 的输入为时间和幅度上均离散的二进制信号，输出为时间连续但幅度上仍然离散的电压值。基本上所有 D/A 都使用寄存器来接收数字系统的输出值，在下一个值收到之前，该寄存器将保持当前电压值。不同于 A/D，D/A 不是串行系统，转换速率可以很快，因而输出类似于阶梯电压，该过程如图 4.25 所示。

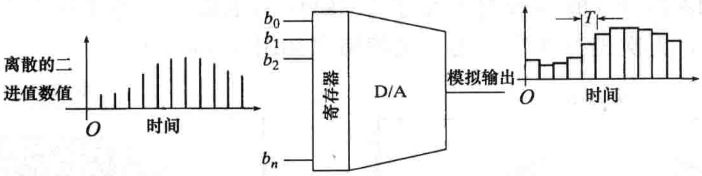


图 4.25　D/A 及其输入/输出信号

4.5.1 D/A 转换技术

最容易理解的 D/A 为加权电阻网络转换器，设二进制数字 Y_B，共有 B 位，表达式为 $Y_B=b_{B-1}b_{B-2}b_{B-3}\cdots b_1b_0$，其中，$b_i$ 是二进制数(1 或 0)。

在二进制中，每位代表相应的 2 的幂次，可以将 Y_B 表示的值写为

$$Y_B = 2^{B-1}b_{B-1} + 2^{B-2}b_{B-2} + \cdots + 2b_1 + b_0$$

二进制数的每一个位均有一个权值，其为 2 的幂次，具体的幂值由其所在二进制数中的位置决定。

在电路中，可以使用离散的电压值表示二进制数的位值，如 0V 等于二进制中的 0，5V 等于二进制中的 1。对于任一个电阻，若该电阻两端的电压为 0V，则产生的电流为 0，若电阻两端的电压为 5V，产生的电流与电阻值成反比。基于这种理念构建 D/A，将电阻值与加权系数相对应，设计电阻值使其与 2 的幂次相吻合，使用运算放大器将电流相加。该 D/A 的原理图如图 4.26，其为实际 D/A 的简化图，假设二进制位值为 0V 或 5V 的电压。在实际中，使用这些位的值来控制开关以决定加在电阻器两端的参考电压值。电阻器 R_g 为增益电阻，调整该电阻值，可使输出电压在合适的范围。

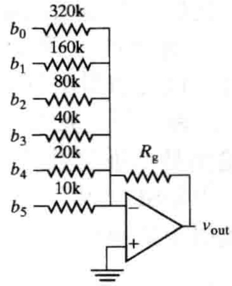


图 4.26　权电阻 D/A 转换器，在该转换器中，电阻为加权系数，运算放大器将电流相加

不同于模拟数字转换，D/A 转换不需要预测、统计，以及其他的串行处理过程，转换速率由信号通过元件的时间决定，同时也不需要时钟。

D/A 转换器在实际应用中的一个主要问题如图 4.26 所示。当转换位数很大时，电阻值的设计变得很困难。例如，对于 10 位转换器，如果 b_9 位对应的电阻值为 10kΩ，则 b_0 对应的电阻值为 5.12MΩ。对于 12 位转换器，若 b_{11} 对应的电阻值为 10kΩ，则 b_0 对应的电阻值将为 20.48MΩ。对于阻值较大电阻器，要求其具有很高的精度，例如 10 位的转换器可以识别的数值为 $1/2^{10}$，即 0.1%，若电阻器的阻值范围较宽且不进行调整，则无法保证如此高的精度。

为克服多值电阻器的精度问题，另一个实现 D/A 功能的电路如图 4.27 所示。该转换器称为梯形电阻网络，也称为 $R/2R$ 电阻网络转换器，其仅使用两个不同的电阻值。在构建集成电路时，构建两个基本相同的电阻（两电阻的电阻值相差在 1%之间）比构建两个相对于某一特定值为 1%误差的电阻（两个电阻的阻值不相同）更容易。

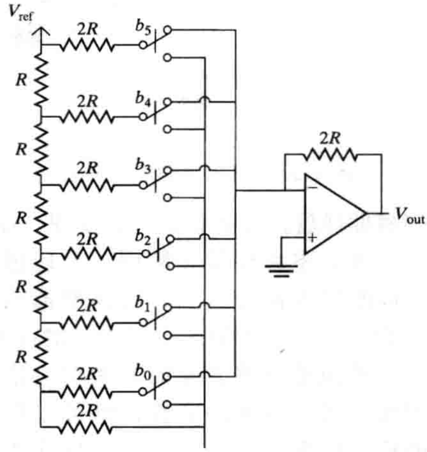


图 4.27　6 位梯形 D/A，注意该电路只使用 2 个不同阻值的电阻，二进制数值 b_0～b_5 用于控制开关

在图 4.27 中，使用 b_0～b_5 控制的电子开关替代机械开关。由于分压网络的设计，到达任一个开关的电流均为到达上一个开关电流的一半。

在实际应用中，D/A 转换时间一般都很快，因此转换速度并不是设计信号处理单元考虑的重要因素。音频系统中常用 16 位转换器，在市场上很容易买到。另外根据价格和应用的不同，典型的 10～12 位 D/A 的转换时间大约在 40ns～10μs 之间，更高速度的 D/A 可以达到 1GSPS。

4.5.2　精度和误差源

数字模拟静态转换误差的定义与模拟数字转换中的静态误差相似，包含偏移误差、增益误差、微分非线性误差和积分非线性误差。

D/A 中信噪比的定义也与 A/D 中的定义相似，也可使用式(4.4)计算。然而，D/A 中的动态误差不同于 A/D 中的动态误差。对于 D/A 来说，设计者和最终用户需考虑转换器的瞬态响应，瞬态响应由毛刺和电压转换速率两个因素所决定。

毛刺是专业术语，在 D/A 中，其表示开关从一个状态转换到另一个状态的不确定性引起的输入信号改变，进而在输出端产生异常。例如，如果 D/A 的输入由 1000 1000 转换到下一个状态 1000 0111，最后面 4 位的状态同时改变。在 D/A 内部的开关中，某些开关的速度可能会比另一些开关速度稍快，因而输出端就会产生毛刺，其某种程度上类似于逻辑电路的竞争现象。为避免毛刺产生，某些 D/A 在其输出端增加抗毛刺电路，类似于 A/D前端的抽样保持电路。

转换速率是衡量当输入出现一个跳变时，输出产生相应改变的响应时间。如果转换速率较低，在输出端会产生谐波失真。对于正弦信号，最大的斜率发生在 $k\pi(k=0, 1, 2, \cdots)$的位置，该处的斜率值为$\pm 2\pi f$，其中 f 为以赫兹表示的正弦信号频率。因此，对于$y=A\sin(\omega_0 t)$表示的正弦波，为防止谐波失真的产生，转换速率需$\geqslant A\omega_0$。

DSP 的输出是二进制数值，该数值进入输出寄存器，该寄存器作用于 D/A。在抽样周期结束时，由 DSP 产生一个新的数值来替换旧的数值，D/A 也完成一次转换。这些离散的输出值类似于阶梯的形状，如图 4.22 所示。

4.6 抗镜像滤波器

信号通过理想抽样器后，信号频谱和原始信号频谱是相同的，只是抽样信号的频谱在频域中呈周期性出现，其周期由抽样频率决定，如图 4.28 所示。

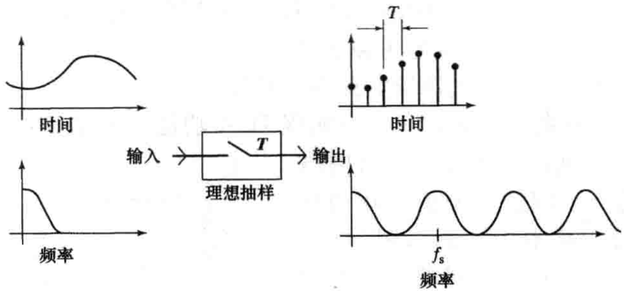


图 4.28 该图为理想抽样器输入和输出信号的时域和频域，抽样频率 $f_s=1/T$

观察抽样后信号的频谱可以发现，如果去除这些频谱中的重复镜像部分，即可恢复原始信号。实现上述目标需要使用一个理想低通滤波器，该滤波器在 $0\sim f_s/2$ 的频率范围内增益为 1，在高于 $f_s/2$ 的频率范围内，增益为 0。

由于理想低通滤波器的冲激响应是非因果的，因而在实际中无法实现，但可以得到该滤波器的近似模型。实际上，D/A 转换器使用寄存器来保持两个抽样点间 DSP 的输出，进而产生阶梯效应，该系统在某种程度上达到了低通滤波的效果。

寄存器- D/A 等效的频率响应函数如图 4.29 所示(如何得到该函数请参考习题 4.11)，在很多实际应用中，这就是所需的滤波效果。例如，阶梯波的输出形式很适合控制直流电机。对于仅使

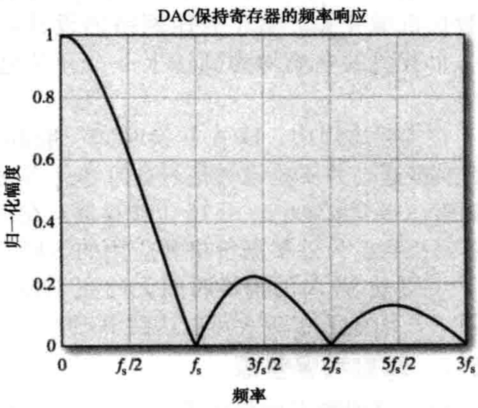


图 4.29 产生阶梯输出的寄存器和 D/A 的频率响应函数

用保持寄存器而不能满足要求的情况，需要设计更复杂的输出电路。

在某些系统中，在信号进入到 D/A 之前需要在 $0\sim f_s/2$ 的频带内，增加信号的增益以补偿衰减。或者也可在 D/A 之后，以模拟滤波器的形式增加补偿。此外，也可使用不同类型的转换器，如 Σ-Δ 转换器。◀

例 4-6 抽样频率为 f_s 赫兹的信号通过 D/A，然后进入保持寄存器，在信号进入 D/A 之前需进行何种处理以补偿保持寄存器的衰减？

解：图 4.29 给出了保持寄存器滤波效果，为补偿其滤波效果，需要在信号进入 D/A 之前加入增益，以使增益与滤波器幅频响应的乘积在 $0\sim f_s/2$ 范围内为 1，保持滤波器的滤波曲线表示为

$$H(\omega)=(1-\mathrm{e}^{-\mathrm{j}\omega T})/\mathrm{j}\omega T$$

为得到补偿滤波器的频率响应，需进行如下处理：

```
fs = 1000;T = 1/fs;
f = 1:fs/2;w = 2*pi*f;
H = (1 - exp(-j*w*T))./(j*w*T);
Ideal = ones(1, length(H));
figure(1);clf;
%Normalize magnitude
H = abs(H)/max(abs(H));
plot(f, H);
hold on;
%Divide normalized magnitude into 1 to get compensator
Hc = Ideal./abs(H);
plot(f, Hc);
xlabel('Frequency');
ylabel('Normalized magnitude');
title('Response of the D/A and hold register');
```

图 4.30 为结果。◀

在 DSP 的前端使用模拟低通滤波器，在后端也使用模拟低通滤波器。前端的滤波器用于抑制信号中大于 $f_s/2$ 的频率分量，后端的滤波器用于去除抽样过程引入的额外的频率分量，也称为重建滤波器。

在某些应用中，对输出信号过抽样可以降低代价，同时可以提高模拟重建滤波器的质量。考虑简单的 RC 低通滤波器，其输入为抽样后的信号，使用电容存储抽样点的电量，使用该电量来填充抽样点间的空白以平滑信号。如果抽样频率增加，抽样点间的距离将变小，则电容器尺寸将降低，因此过抽样可以降低电路的成本和尺寸。

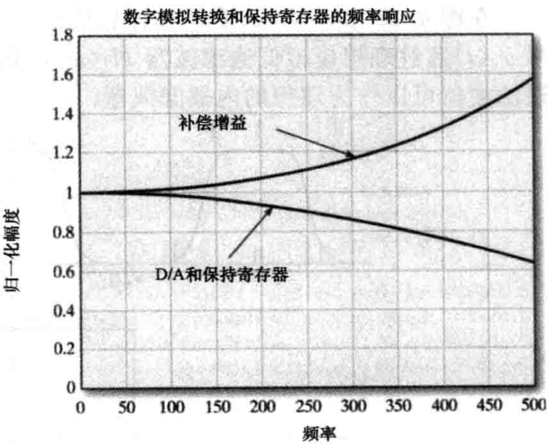


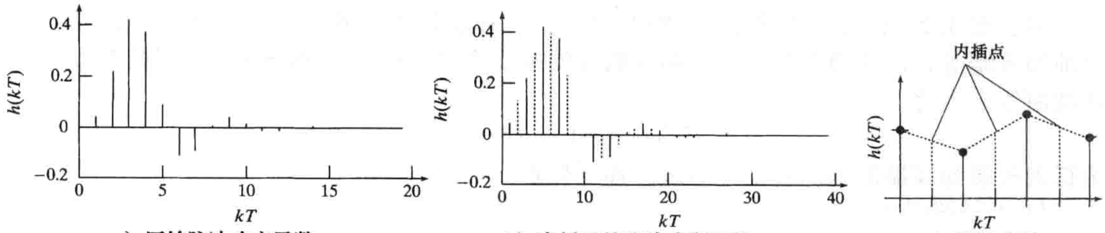
图 4.30 在信号进入到 D/A 和保持寄存器前加入补偿

对于音频系统，抽样频率一般为几十千赫兹，增加抽样频率到几百千赫兹，则模拟重建滤波器的尺寸将降低。对于给定的信号，增加其抽样频率的过程称为过抽样或内插，新插入的抽样点值来源于其邻近的抽样值的加权平均近似。

内插滤波器在 CD 和某些控制系统中有广泛的应用，CD 播放器一般采用 4 倍过抽样，

即在信号进入到扬声器前，使用内插滤波器，这样感觉存储在磁盘中的信号就像由4倍抽样频率得到的信号。类似地在控制系统中，当遇到低频控制信号的情况时，在控制信号输入到连续时间电机前，常使用易于通过计算实现的内插滤波器，否则需要使用成本较高的高质量模拟滤波器。

线性内插是一种简单且容易理解的内插方式，图4.31中是将线性内插应用到脉冲响应函数的情况。



a）原始脉冲响应函数　b）内插后的脉冲响应函数，注意抽样速率变为原来的2倍　c）线性内插

图4.31　线性内插

对于实际的线性内插器，一般先在抽样点间插入零值，如图4.32所示。内插滤波器使用前一时刻抽样点值和下一时刻抽样值的平均值替代零值，由于滤波器不能获取未来时刻的信号，在输入序列中需要引入单位延时器。

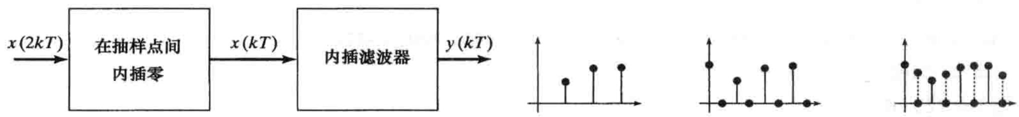


图4.32　线性内插滤波器的实现

尽管线性内插滤波器很容易实现，但其效果并不总是让人满意，实际上，在频域中能够得到更好的解决方法。

在图4.33中，$X_s(\omega)$表示抽样后信号$x_s(t)$的频谱函数，该函数为周期函数。如果信号$x_s(t)$通过理想模拟低通滤波器$H_I(\omega)$，则仅保留信号一个周期内的频谱，理想模拟低通滤波器可以作为理想的内插滤波器。

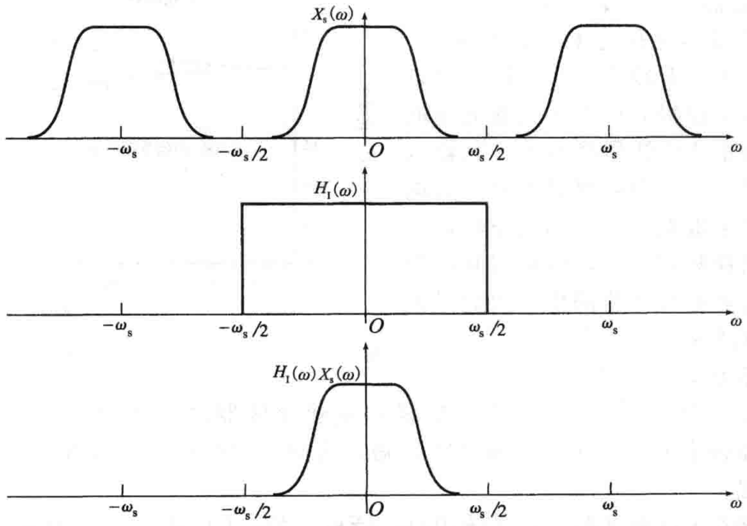


图4.33　当抽样后信号频谱通过理想低通滤波器时，可以恢复原始信号频谱

在时域中，理想模拟低通滤波器的冲激响应为

$$h_{理想}(t)=\frac{1}{2\pi}\int_{-\infty}^{\infty}H_I(\omega)\mathrm{e}^{\mathrm{j}\omega t}\mathrm{d}\omega=\frac{1}{2\pi}\int_{-\omega_s/2}^{\omega_s/2}(1)\mathrm{e}^{\mathrm{j}\omega t}\mathrm{d}\omega$$

$$h_{理想}(t)=\frac{1}{2\pi \mathrm{j}t}\mathrm{e}^{\mathrm{j}\omega t}\Big|_{-\omega_S/2}^{\omega_S/2}=\frac{\mathrm{e}^{\mathrm{j}\omega_s t/2}-\mathrm{e}^{-\mathrm{j}\omega_s t/2}}{2\pi \mathrm{j}t}=\frac{2\mathrm{j}\sin(\omega_S t/2)}{2\pi \mathrm{j}t}=\frac{\omega_S}{2\pi}\mathrm{sinc}(\omega_S t/2) \quad (4.5)$$

在频域中，理想滤波器和抽样后信号的频谱相乘，频域乘积对应于时域卷积，因此，在时域中，信号 $x_s(t)$通过内插器的响应 $y_{\mathrm{int}}(t)$可以写成

$$y_{\mathrm{int}}(t)=x_s(t)*\frac{\omega_S}{2\pi}\mathrm{sinc}(\omega_S t/2)$$

或

$$y_{\mathrm{int}}(t)=\frac{\omega_S}{2\pi}\int_{-\infty}^{\infty}h(\tau)\mathrm{sinc}[\omega_S(t-\tau)/2]\mathrm{d}\tau$$

以离散时间函数表示，该方程可以写成

$$y_{\mathrm{int}}(t)\big|_{t=kT}=h_{\mathrm{int}}(kT)=\frac{\omega_S}{2\pi}\sum_{k=-\infty}^{\infty}h(\tau)\mathrm{sinc}[\omega_S(kT-\tau)/2] \quad (4.6)$$

从式(4.6)可以看出，内插方程可以解释为多个延时的 sinc 函数的加权和。求和的范围是无限且非因果的，因此理想的内插滤波器在现实中无法实现。但可以使用过渡带较窄的低通滤波器近似理想的内插滤波器。在实际应用中，对于音频系统来说，相位对信号的影响较大，因此使用 FIR 滤波器比较合适；在控制系统中，IIR 滤波器计算效率很高，因此更合适。第 8 章将对过抽样和内插滤波器进行更详细的介绍。

习题

概念题

1. 假设在 A/D 转换过程中没有使用抽样保持电路，将可能导致哪种误差？
2. 在什么情况下，可以不使用抽样保持电路？
3. 若设计阻带衰减为 70db 的低通滤波器，需要多少位的 A/D 转换器？
4. 某数字滤波器由抽样保持电路、A/D、计算系统和 D/A 构成，可以使用示波器观察输出结果，且可使用不同频率的正弦信号作为输入，设计两种方法以测定该滤波器的抽样频率。
5. 假设抽样保持电路有较大的电压衰减，比如 10%，该误差对于输出结果有何影响？同时考虑幅度响应和相位响应。
6. 为什么在闪存转换器中一般不需要抽样保持电路？
7. CD 播放器中过抽样滤波器的作用是什么？为什么需要使用该滤波器？如果没有过抽样滤波器，为了能够在输出端得到相同的音频质量，可使用什么替代方法？
8. 对于相同的位数，为什么 D/A 转换速率通常要比 A/D 转换速率总体上快将近 10 倍？
9. 当我们看电影时，经常会发现尽管汽车向前移动，但感觉到汽车的轮子向后移动，使用混叠来解释这种现象。抽样的是什么信号？混叠是怎么产生的？
10. 如果对信号过抽样，可以使用低阶的抗混叠滤波器替代高阶的抗混叠滤波器，过抽样过程可看成是将抗混叠滤波器的设计从硬件问题转为软件问题，解释为什么这样说是正确的。
11. 在讨论 A/D 中的误差时，介绍了 4 种类型的误差：偏移误差、增益误差、微分非线性误差和积分非线性误差。现有一个 A/D 转换器和一个信号发生器，通过观测 A/D 的输入和输出以测量每一种误差。
12. 对于一个频率为 400Hz 的正弦信号，以 5000Hz 进行抽样，通过每 10 个点去除 9 个点对该正弦信号进行 10 倍降抽样，则降抽样后的正弦信号频率将为 100Hz。解释其中的原理。
13. 图 P4.13 中的 MATLAB 代码产生了 5000Hz 抽样的 400Hz 正弦信号，通过在抽样点间插零对信号进行升抽样，该代码同时画出了升抽样后信号的频率响应(见图 P4.13)，频谱中显示出原始信号在

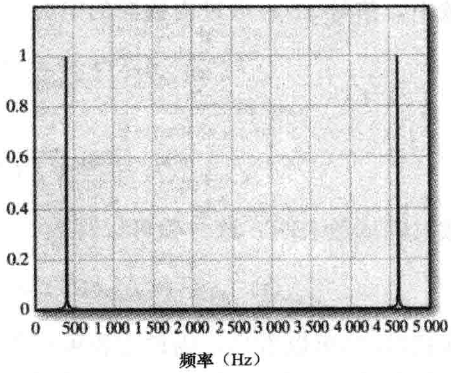
400Hz 位置，同时在 4600Hz 处有第二个信号。

(a) 利用混叠解释 4600Hz 信号的产生。

(b) 根据(a)的结果，假设信号频率为 200Hz 或者 2000Hz，则升抽样后的信号频谱将变为怎样？

```
fs = 5000;T = 1/fs;
fsig = 400;
t = 0:T:1000*T;
y = sin(2*pi*fsig*t);
y2 = unsample(y,2);
t2 = 0:T/2:1000*T;
figure(1);clf;
yFFt = fft(y2);
L = length(yFFt);
k = 1:L;
yFFTmag = abs(yFFT)/max(abs(yFFt));
plot(k*2*fs/L, yFFtmag);
axis([0 fs 0 1.2]);
xlabel('frequency in HZ');
ylabel('gain');
title('FFT of signal');
```

a)



b)

图 P4.13 MATLAB 代码和升抽样后的正弦信号频谱

14. 某差分方程对应的单位阶跃响应为 $y(nT)\mid_{x=\text{阶跃信号}}=\{0.4, 1.0, 1.18, 1.09, 0.991, 0.97, 0.987, 1.003, 1.005, 1.00, 1.00, 1.00, \cdots\}$

仅通过该阶跃响应，关于该差分方程对应的频率响应，可以得出什么结论？

15. 为了保持待抽样信号的信息，需要以至少 2 倍的信号最高频率对连续时间信号进行抽样。该抽样定理的表述有什么错误？

16. 车轮中有从中心到边缘的辐射线，当使用每分钟闪烁 100 次的闪光灯照射车轮时，车轮看起来是不动的。从该现象推断，该旋转的车轮其转速为多少？

分析设计题

4.1 节

4.1 以 44100Hz 抽样 1000Hz 的正弦信号，计算该信号的前三个混叠频率值。

4.2 信号被抽样后，依次通过 A/D、滤波器和 D/A，在没有抗混叠滤波器和抗镜像滤波器的情况下，使用示波器观察输入和输出，观察到 f_0 的前两个混叠频率为 f_1、f_2，则 f_0 和 f_1、f_2 之间的关系为什么？

4.3 两个信号 $x_1(t)$和 $x_2(t)$，均为 $f_S/2$ 的带限信号，抽样频率为 f_S，找到如下信号的最小抽样频率，(a) $x_1(kt)$，其中，k 为正整数，(b) $x_1(t/k)$，(c) $x_1(t)*x_2(t)$，其中，$*$ 表示卷积。

4.4 信号 $f(t)=4\sin(200\pi t+\pi/6)-3.2\cos(300\pi t-\pi/6)$，其奈奎斯特频率是多少？

4.5 抽样器、A/D 和 D/A 级联使用，使得输出为输入的抽样结果。假设抽样频率为 1000Hz，输入信号 $x(t)=\sin(200\pi t)$，输出将为 $y(nT)=\sin(200\pi nT)$，其中，T 为抽样周期，找到另外两个与该信号产生相同输出的连续时间信号。

4.6 在第 3 章中，将占空比设为 50%，频率为 f_0 的方波信号的傅里叶级数写成

$$f(kT)=\sum_{\substack{k=1\\k\text{为奇数}}}^{N}\frac{4}{k\pi}\sin(2\pi f_0 kT)$$

写出该方波的奈奎斯特频率，以 N 的函数表示。

4.7 以 1000Hz 抽样 800Hz 的信号，将输出结果输入到 D/A，该输出信号的基本频率为多少？在输出结果中，每个周期中有多少个抽样点？

4.8 以 1000Hz 为抽样频率，对于频率分别为 50Hz 和 0.001Hz 的正弦信号，计算其相邻抽样点间的最大电压差值。对于低频率(相对于抽样频率)输入信号，有效位个数的含义是什么？

4.9　信号的频谱 $X(\omega)$ 如图 P4.9 所示，抽样频率为 ω_s，带限信号的带宽为 ω_B，该图中，$\omega_B=2\omega_s/3$，简要画出抽样后信号的频谱图。

4.2 节

4.10　假设使用 3 阶的模拟巴特沃斯滤波器作为抗混叠滤波器，该滤波器的截止频率为 11025Hz，系统的抽样频率为 22050Hz。假设系统中的噪声频率为 12127.5Hz，幅度为 0.5，则噪声经过系统后有什么变化？

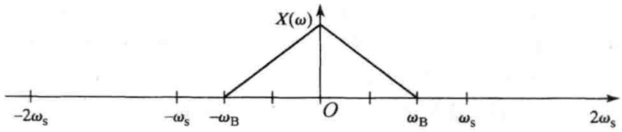


图 P4.9　画出抽样后信号的频谱

4.3 节

4.11　在例 4-3 中，使用 MATLAB 程序得到了量化位数为 10 位、电压范围在 −1～+1V 之间的正弦信号的 QSNR。若正弦信号的幅度范围在 −0.4～+0.4V 之间，则 QSNR 是多少？

4.12　修改例 4-3 中的 MATLAB 代码，使得位数 b 的范围在 1～24 之间，画出 QSNR 与 b 的关系曲线。

4.13　修改例 4-3 中的 MATLAB 代码，得到下列信号的 QSNR。(a) 信号 x 为方波信号，基频为 100Hz，使用傅里叶级数的前 10 个谐波拟合该方波，参考附录 D 中的 fouriersq 函数。(b) 信号 x 为三角波信号，基频为 100Hz，使用傅里叶级数的前 10 个谐波拟合该三角波，参考附录 D 中的 fouriertri 函数。

4.14　在式(4.4)中，如果信号为正弦信号，QSNR 为 QSNR≈6.02b+1.76。

如果输入不是正弦信号，则写成 $\mathrm{QSNR}\approx 10\lg(P_S^2/(\Delta/12)^2)$，其中 $\Delta=x_{\max}/2^b$。

证明上式可简化为 $\mathrm{QSNR}\approx 6.02b+10.81-20\lg(x_{\max}/P_S)$。

4.4 节

4.15　使用图 4.15 所示的 SAR 结构构建一个 10 位 A/D 转换器。其中 10 位 D/A 转换器的转换时间为 1μs，反馈通道的逻辑控制单元每 50ns 进行一次电压预测。

(a) 估计完成 10 位转换所需的时间；

(b) 假设仅需 8 位转换，则转换时间是否有变化？如果有变化，则 8 位转换所需的时间为多少？

4.16　使用 8 位 A/D 对温度传感器的信号进行转换，传感器的误差为±3%，电压范围在 0～8V 之间，A/D 转换器的满量程动态范围为 0～10V 之间。如果考虑量化误差，则 A/D 转换器输出的数字信号的精度为多少？

4.17　A/D 的 ENOB 为 13.5 比特，如果 A/D 前的放大器电压误差为 0.1%，抽样保持电路的电压误差为 0.3%，如果输入为理想的正弦信号，则 A/D 输出的噪声水平为多少，以分贝表示。

4.18　传感器的输出电压信号范围为 0～10V，误差为±3%，为其选择一个 A/D 转换器，为使在最坏的情况下，总体误差不超过±3.1%，则 A/D 所需的最少位数为多少？

4.19　传感器的输出电压为 0～5V，10 位 A/D 的输入电压范围为 −10～+10V，如果使用 b 位，输入电压范围在 0～5V 的转换器替代该 10 位 A/D。为达到与该 10 位 A/D 转换器相同的精度，b 的最小值应为多少？

4.20　系统的抽样频率为 22050Hz，如果使用 2 倍过抽样，则模拟滤波器的通带和阻带为多少？如果使用 4 倍过抽样，则通带和阻带为多少？

4.21　过抽样可以降低对模拟抗混叠滤波器的要求，但是并没有去除模拟抗混叠滤波器。假设抗混叠滤波器的通带波动为 $\pm R_{pa}$，数字抗混叠滤波器的通带波动为 $\pm R_{pb}$，将两个滤波器结合使用，则通带波动为多少？

4.22　A/D 中的时钟抖动会产生噪声，该噪声正比于抖动的幅度以及信号的频率。

(a) 在 MATLAB 中，为了描述抖动的幅度与噪声之间的关系，抽样频率、抽样周期和理想正弦信号如下：

```
fs = 11025;T = 1/fs;
t = 0:T:0.01;
fsig = 1000;
gIdeal = sin(2*pi*fsig*t);
```

为了加入抖动，可生成一个幅度为 k 的随机变量 r，并将其加到 t 中，如下所示：

```
r = k*(rand(size(t) - .5)*T;
g = sin(2*pi*fsig*(t+r));
```

k 的范围为 0～1，抖动噪声正比于理想正弦信号 gIdeal 和 g 之间差值的均方根，画出抖动噪声和 k 的关系。

(b) 令 $k=0.5$，使输入信号的频率范围为 100Hz～$f_S/2$，重复(a)。

4.23 如习题 4.22，创建抖动幅度 $k=0.1$ 的抖动信号和正弦信号，正弦信号的频率为 $f_S/4$Hz，抽样频率 $f_S=11025$Hz，产生一段时间内(至少持续 100 个周期)的正弦信号，以分贝形式画出 0～$f_S/2$Hz范围内加入抖动后信号的频谱图。

4.24 图 4.20 所示的 Σ-Δ 型 A/D 转换器可以简化为图 P4.24 中的形式，使用拟合积分系统来替代积分器，该系统使用宽度为 T 的矩形累加拟合积分。假设输入为正弦信号，使用 MATLAB 模拟转换器，产生调制后的输出信号 $y_1(t)$。

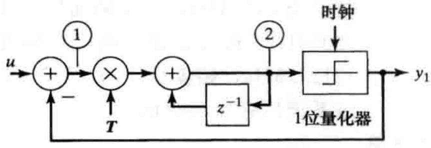


图 P4.24 使用矩形积分器的 Σ-Δ 转换器

(a) 验证可以使用矩形累加进行拟合节点 1 和 2 之间的积分器。

(b) 节点 1 处的信号为 $n_1=u-y_1$，使用 u 和 y_1 的差分方程形式，写出节点 2 处的信号。

(c) 使用 MATLAB 创建输入信号 u，该信号为 100Hz 的正弦信号，抽样频率 $f_S=44100$Hz，信号的持续时间至少为 0.1 秒。编写循环程序，找到输出信号 y_1 与输入信号 u 之间的关系，同时画出输出信号与时间的关系图。

4.25 在习题 4.24 中，使用矩形积分近似 Σ-Δ 转换器中的积分器，也可以使用梯形来近似该积分器，如在双线性变换(BiLinear Transform，BLT)的情况。使用梯形近似积分器的传输函数为

$$H_1(z)=\frac{T}{2}\frac{z+1}{z-1}$$

求解梯形积分器的差分方程，重复习题 4.24 中的(c)。

4.26 逐次逼近寄存器模拟数字转换技术如图 4.15 所示，设 n 位数值的最高有效位为 b_{n-1}，使用如下代码模拟逐次逼近算法：

```
Set n = number of bits
k = n - 1
initialize input;
set guess to 0
while(k >= 0)
guess = guess + 2^k
if(guess > in)
guess = guess - 2^k
end
k = k - 1;
end
```

利用 MATLAB 实现该代码，当输入值为 1023 且 $n=10$ 位时，画出预测值与计数器 k 之间的关系。

4.27 如图 P4.27 所示，将两个闪存转换器结合使用增加转换位数是可行的，但该原理图也存在一些问题。如果 8 位转换结果的精度为半位(nearest half bit)，则 4 位 D/A 和模拟减法器的精度是多少？

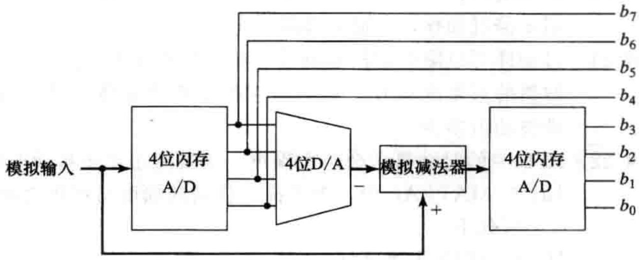


图 P4.27 两个 4 位闪存转换器组合使用形成 8 位转换器

4.5 节

4.28 在 D/A 的输出端增加一个寄存器用来保持当前样点值，其产生的输出结果为阶梯波，写出寄存器的传输函数和频率响应，画出幅度谱与频率的关系。

4.29 输出端带有保持寄存器的 D/A 等效地使输出信号通过了频率响应如图 P4.29 所示的低通滤波器。如果理想低通滤波器在 $0\sim f_s/2$ 范围内的增益为 1，其他频率处增益为 0。简要画出补偿滤波器的频率响应，以使得该补偿滤波器与保持寄存器级联使用时得到理想滤波器的频率响应。

4.30 假设 10 位 A/D 的输入信号范围在 $-3\sim+3$V 之间，该转换器将 -3V 换为 00 0000 0000，如果输入信号为 0.514 伏，则该 A/D 的输出是什么？量化误差为多少？

4.31 假设两个完全相同的低通滤波器，其频率响应如图 P4.31 所示，将两个滤波器以串联方式级联使用，若将频率为 250Hz，幅度为 3.0 的正弦信号输入第一个滤波器，求解该级联系统的稳态输出。

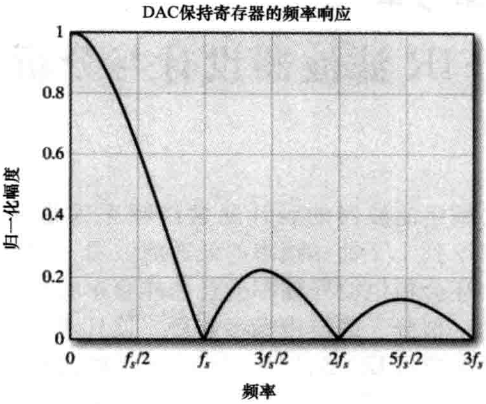


图 P4.29 输出阶梯波的 D/A 寄存器的频率响应函数

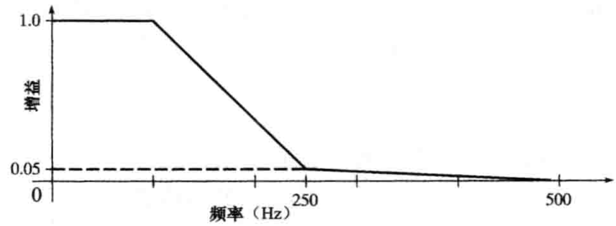


图 P4.31 低通滤波器的增益与频率关系

第5章

FIR 滤波器设计与分析

数字滤波器的设计通常以频率为指标进行。例如，设计通带为 0～200Hz，阻带为 225Hz 到 1/2 抽样频率的低通滤波器。设计指标一般还指明通带与阻带内的波动参数。此外，还会指定抽样频率或者抽样频率的范围，在这种情况下，选择抽样频率也是滤波器设计的一部分。确定指标参数后，设计者必须选择使用无限冲激响应(IIR)滤波器还是有限脉冲响应(FIR)滤波器。针对这两种选择，还有更多的选项需要考虑。例如，IIR 滤波器中有巴特沃斯、切比雪夫等类型，FIR 滤波器可以使用帕克斯-麦克莱伦(Parks-McClellan)算法、频率抽样法或者窗函数法。设计结果通常是给出 z 域传输函数。如果滤波器设计之后直接实现，设计质量上往往会出现问题，因此在实现滤波器之前，设计者需要对设计的滤波器进行分析以确定它是否满足设计指标。设计过程应是交互的，一个好的设计者一般会尝试多种方案，然后进行分析比较，最后选择最佳的设计结果进行实现。即使在实现之后，进一步分析也可以优化设计方案。

分析在滤波器设计过程中发挥着至关重要的作用。通过分析不仅可以知道设计指标是否圆满地达到了，而且有利于设计者做出明智的取舍以改进设计。

数字滤波器的分析集中在传输函数。传输函数用于描述实现之前的滤波器，如果确定了传输函数，那么我们可以从数学上确定所期望的结果(假如滤波器在理想情况下实现)。依据传输函数可以确定滤波器的脉冲响应、稳定性、稳态的频率响应、差分方程以及任意输入的响应。

滤波器分析可以分为两部分：瞬态分析与频率分析。通过瞬态分析可以判断滤波器是否稳定。更重要的是，它提供了一种可以确定滤波器相对稳定性的方法。例如，一个滤波器输入阶跃响应，通过瞬态分析可知过冲和波动的强弱。频率分析可以确定所设计的滤波器类型以及如何满足设计指标。频率分析是滤波器对正弦输入信号的响应，需要强调的是，频率分析依据的是正弦*稳态响应*，而不包括正弦信号开始输入时产生的瞬态响应。

本章简述了 FIR 数字滤波器设计与分析的基本概念。作为一种设计技术，我们应用傅里叶级数以及窗函数，并详细介绍线性相位响应的条件。我们从时域与频域分析了 7 种窗函数：矩形窗、三角窗、汉明窗、汉宁窗、布莱克曼窗、凯瑟窗和道尔夫-切比雪夫窗，同时利用图形卷积解释它们的内部工作机理。最后讨论最佳的 FIR 滤波器设计方法——帕克斯-麦克莱伦算法及其应用。

5.1 滤波器设计指标

滤波器通常在频域内指定设计指标，时域指标却很少见。图 5.1 给出了典型的 FIR 低通滤波器频域参数。

图 5.1 所示的是一个线性指标图，而波动与衰减通常以分贝指定。分贝是一种用来表征声响或声压的计量单位，在信号处理与通信系统中，常用来衡量信号的强弱。分贝计算公式为 $db=10\lg(P_{out}/P_{in})$。由于功率正比于电压的平方，所以电压分贝定义为 $db=20\lg(V_{out}/V_{in})$。阻带衰减以分贝定义为 $-20\lg(R_s)$，称为阻带的分贝波动参数。通带范围内，

滤波器增益归一化为 1，通带波动可以大于 1 也可以小于 1，如图所示。通带波动以分贝定义为 $-20\lg(1-R_p)$。

例 5-1　分贝波动

假设希望设计一个数字滤波器，其通带与阻带波动都是 0.01，那么通带与阻带的分贝波动各是多少？

解：通带波动为：

$$-20\lg(1-R_p)=-20\lg(0.99)=0.0873\text{dB}$$

阻带波动为：

$$-20\lg R_s=-20\lg(0.01)=40\text{dB}$$ ◀

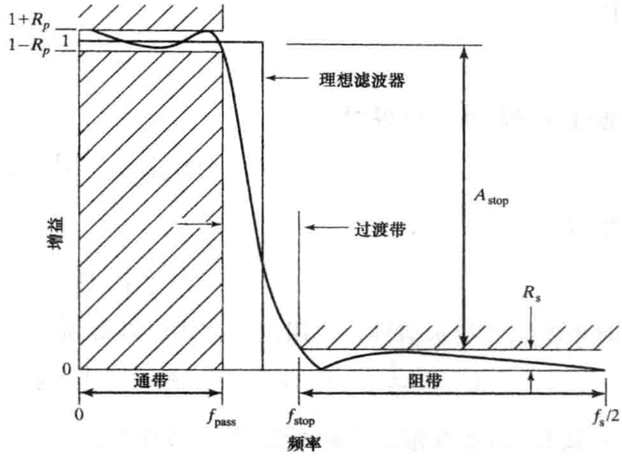


图 5.1　数字滤波器在频域的参数

R——波动或者给定范围内可接纳的过冲与下冲量。

R_p——通带波动，对于低通滤波器，通带增益被归一化为 1，通带最大增益为 $1+R_p$，最小增益为 $1-R_p$。

R_s——阻带波动，对于低通滤波器，阻带最大增益为 R_s。

A_{stop}——阻带衰减。此参数通常使用分贝表示，它与阻带波动 $A_{stop}=-20\lg(R_s)$有关。

f_{pass}——低通滤波器的通带最高边界频率。

f_{stop}——低通滤波器的阻带最低边界频率。

$f_{stop}-f_{pass}$——过渡带带宽。

5.2　FIR 滤波器设计的基本原理

FIR 滤波器设计存在多种重要方法，本节将讨论的最小二乘设计法，并不是始于某一个频率参数指标而是依赖于理想滤波器。该方法有一个固有的难题，即无法提前预测滤波器的阶数，在满足所需设计指标之前，设计者往往要尝试多种可能性。本节最后讨论的凯瑟窗(Kaiser window)和帕克斯-麦克莱伦算法可以基本解决这个问题。

暂且偏离本节主题回顾一下傅里叶级数。在表示如方波这样的周期函数时，可以使用无限个正弦函数之和的方式。重要的是，只有无限项序列的傅里叶级数才能确切地表示该函数。如果只有有限项表示，数字滤波器恰好就是这种情形，这种表示只是近似的估计。可以建立一个误差函数，通过计算真实函数与傅里叶级数表示之间的误差来确定估计的好坏。如果傅里叶级数限定为 $2N+1$ 项，就称此估计为 $F_N(t)$：

$$F_N(t)=\frac{a_0}{2}+\sum_{k=1}^{N}a_k\cdot\cos(\omega_0kt)+\sum_{k=1}^{N}b_k\cdot\sin(\omega_0kt)=\sum_{k=-N}^{N}C_k\cdot e^{jk\omega_0t}$$

由式(3.5)和式(3.14)可知，如果使用有限项的傅里叶级数估计周期函数，最小均方误差是该项数的最小化。使用该信息设计 FIR 滤波器，回顾抽样定理，时域抽样导致频域的周期化，反之亦然。时间上离散的理想低通滤波器的频率响应近似如图 5.2 所示。

如果图 5.2 所示为时域信号，则可在频域求得其对应的傅里叶级数。因为它已经在频域中，可以做如下变量转换：

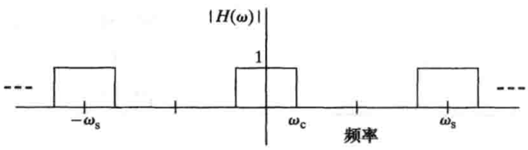


图 5.2　时域抽样的理想低通滤波器

令 $t\leftarrow f$ 以便 $T_0\leftarrow f_s$。

傅里叶级数的指数形式为：

$$x(t)=\sum_{k=-\infty}^{\infty}C_ke^{jk\omega_0t}$$

其中

$$C_k = \frac{1}{T_0}\int_{T_0} x(t)\mathrm{e}^{-\mathrm{j}k\omega_0 t}\mathrm{d}t$$

且

$$\omega_0 = \frac{2\pi}{T_0}$$

通过变量转换，可得到

$$H(\omega) = \sum_{k=-\infty}^{\infty} C_k \mathrm{e}^{\mathrm{j}k2\pi f/f_s}$$

其中

$$C_k = \frac{1}{f_s}\int_{f_s} H(\omega)\mathrm{e}^{-\mathrm{j}k2\pi f/f_s}\mathrm{d}f$$

对于图5.2所示信号，应用以上公式可以得到：

$$C_k = \frac{1}{f_s}\int_{-f_c}^{f_c}(1)\mathrm{e}^{-\mathrm{j}2\pi kf/f_s}\mathrm{d}f = \frac{1}{-2\pi k\mathrm{j}}\mathrm{e}^{-\mathrm{j}2\pi kf/f_s}\Big|_{-f_c}^{f_c} = \frac{1}{k\pi}\sin(2\pi kf_c/f_s) \tag{5.1}$$

如果 $H(\omega)$被看作是传输函数 $\mathrm{H}(z)$在单位圆上的求值，可以进行 $\mathrm{e}^{\mathrm{j}2\pi f/f_s} \leftarrow z$ 替换，得到 z 域中的传输函数，不过存在一个问题，该传输函数为无限项之和，且对应的系统为非因果系统。

$$H(z) = \sum_{k=-\infty}^{\infty} C_k z^{-k}$$

如果截断无穷级数成 $L=N+1$ 项，并在时间上平移 $N/2$，可得到因果、有限的 z 域传输函数(N 是滤波器阶数，L 是滤波器长度)。

$$H(z) = \sum_{k=0}^{N} C_{k-N/2} z^{-k} \tag{5.2}$$

例 5-2　傅里叶级数方法

选择抽样频率 $f_s=11025\mathrm{Hz}$、截止频率 $f_c=2000\mathrm{Hz}$、阶数 $N=10$(长度为11)的如图5.2所示的理想滤波器，使用傅里叶级数方法求解 z 域传输函数的系数。

解：由式(5.1)可得 $C_k=\frac{1}{k\pi}\sin(2\pi kf_c/f_s)$，当 k 由 -5 到 5 变化时，计算出 C_k 对应的值，如表5.1所示。

表5.1　使用式(5.1)计算的 C_k 的值

k	C_k	k	C_k
−5	−0.0351090	1	0.2892013
−4	−0.0786459	2	0.1208196
−3	−0.0291006	3	−0.0291006
−2	0.1208196	4	−0.0786459
−1	0.2892013	5	−0.0351090
0	0.3628118		

由式(5.2)，FIR滤波器的传输函数为：

$$H(z) = \frac{-0.0351090z^{10} - 0.0786459z^9 - 0.0291006z^8 + \cdots - 0.0786459z - 0.0351090}{z^{10}}$$

其幅度谱与相位谱如图5.3所示。

通过相位曲线可以发现，滤波器传输函数的相位是分段线性的，直接位于单位圆上的零点造成了不连续。

FIR 滤波器傅里叶级数设计技术由 MATLAB 中的 firls 函数实现。下列 MATLAB 代码可产生如图 5.3 所示的频率响应。

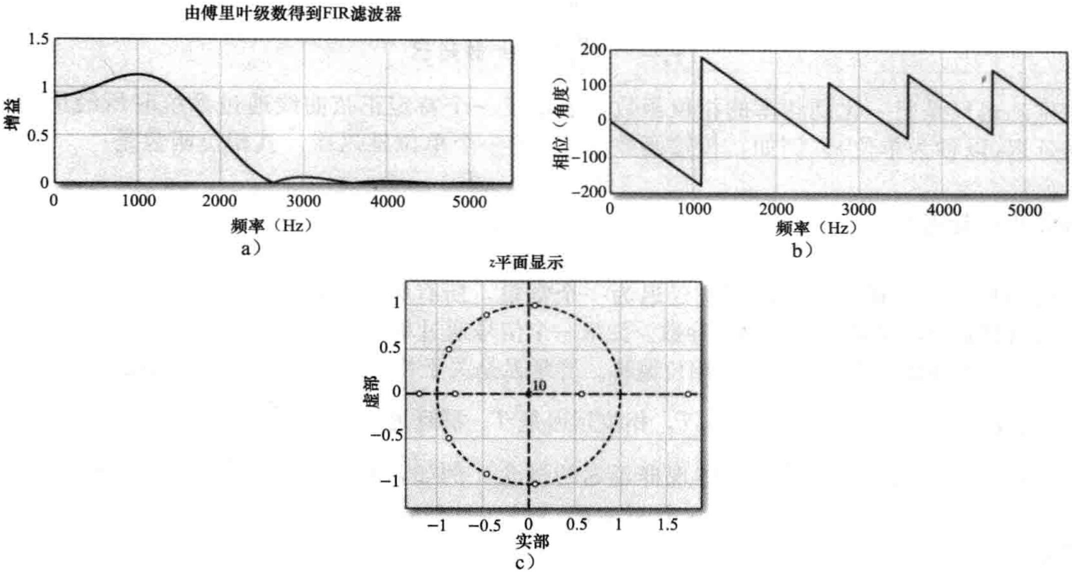


图 5.3　截止频率为 2000Hz 的 FIR 低通滤波器的幅度谱和相位谱

```
N = 10;                                 %Order
fs = 11025;
F = [0 2000/(fs/2) 2000/(fs/2) 1];     %Frequency vector normalized
                                        %to fs/2
M = [1 1 0 0];                          %Corresponding magnitude vector
num = firls(N, F, M);
[H f] = freqz(num, 1, 1024, fs);
figure(1);
subplot(2, 1, 1)
plot(f, abs(H));
axis([0 fs/2 0 1.5]);
xlabel('frequency in Hz');
ylabel('gain');
title('FIR filter from Fourier series');
subplot(2, 1, 2);
plot(f, angle(H)*180/pi);
axis([0 fs/2 -200 200]);
xlabel('frequency in Hz');
ylabel('Phase in degrees');
figure(2);clf;
zplane(num, 1);
title('z-plane plot');
```

5.2.1　线性相位与 FIR 滤波器

一般来说，在满足相同指标的情况下，FIR 滤波器计算效率低于 IIR 滤波器。但是，FIR 滤波器可以具有线性相位，IIR 滤波器只能近似成线性相位，而且还是在一个很小的频带范围。因此在信息被编码成信号相位的应用场合，滤波器不造成信号的相位失真非常重要。

定义两个相位有关的概念

$$\tau_p = -\frac{\theta(\omega)}{\omega} \equiv \text{相位延迟} \tag{5.3}$$

$$\tau_g = -\frac{\mathrm{d}\theta(\omega)}{\mathrm{d}\omega} \equiv \text{群延迟} \tag{5.4}$$

相位延迟只是归一化到频率的相位函数，表示了一个特定正弦曲线通过系统时所经历的时间延迟(以秒为单位)。例如，传输函数为z^{-1}的一个单位延迟器，其相位函数是：

$$\theta(\omega) = -\omega T$$

所以相位延迟为

$$\tau_p = \omega T/\omega = T$$

单位为秒。在这种情况下，相位延迟为一个常量，所有频率都延迟T秒。

群延迟是相位函数对频率的导数。如果一个信号通过单位延迟器，所有频率都将延迟相同时间，但不同频率将会有不同的相位偏移。群延迟是关于相位随频率改变快慢的度量。对于单位延迟器，相位响应是$\theta(\omega)=-\omega T$，相位延迟是T。群延迟$-\frac{\mathrm{d}\theta(\omega)}{\mathrm{d}\omega}=T$也是一个常量。

连续时间调制信号最容易体现群延迟的概念。例如，考虑如下信号：

$$y(t) = \cos(\omega_L t - \phi_L) + \cos(\omega_H t - \phi_H)$$

其中，ω_L略小于ω_H。

由于

$$\cos(a) + \cos(b) = \cos\left(\frac{a+b}{2}\right)\cos\left(\frac{a-b}{2}\right)$$

因此可得

$$\cos(\omega_L t - \phi_L) + \cos(\omega_H t - \phi_H) = \cos\left(\frac{\omega_H - \omega_L}{2}t - \frac{\phi_H - \phi_L}{2}\right)\cos\left(\frac{\omega_H + \omega_L}{2}t - \frac{\phi_H + \phi_L}{2}\right) \tag{5.5}$$

式(5.5)所示的调制波形如图5.4所示。

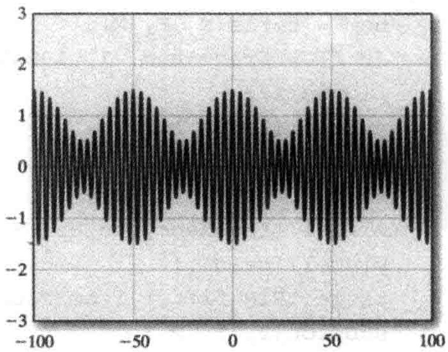


图5.4　式(5.5)所示的调制波形

式(5.5)中的低频余弦信号可以看作包络，而高频信号可以看作载波。对于包络项，提取表示式中的公共因子$\left(\frac{\omega_H - \omega_L}{2}\right)$，可以得到

$$\cos\left(\frac{\omega_H - \omega_L}{2}t - \frac{\phi_H - \phi_L}{2}\right) = \cos\left(\frac{\omega_H - \omega_L}{2}\left(t - \frac{\phi_H - \phi_L}{\omega_H - \omega_L}\right)\right)$$

其中，$\frac{\phi_H - \phi_L}{\omega_H - \omega_L}$可以写成$\frac{\Delta\phi}{\Delta\omega}$，在极限情况下变成$\frac{\mathrm{d}\phi}{\mathrm{d}\omega}$，单位是秒。这种情况下，群延迟成为包络信号的延迟时间。在大多数数字信号处理(DSP)应用中，一般没有载波和简易的包络，但作为相位曲线微分的群延迟，仍然可以很好地衡量相位失真，而且可能是衡量相位曲线变化速率的最好、最简单的方法。

在放大器、扬声器等的设计过程中，往往着重考虑音频信号的相位失真，但是人耳对于相位失真并不敏感，部分原因归结于人耳局限性。音频系统的相位失真往往是十至几百微秒，而人耳只能检测出几毫秒的差异。相位失真在视频系统中对我们越来越重要，相位失真会呈现明显的图像模糊和颜色扭曲。

在通信系统中，相位失真任何时刻都很重要，信息被编码为信号相位或者一个含有高频分量的信号，例如方波，传输过程中的相位失真将会导致信号的离差。

尽管可以近似或者在一些情况下通过修正使得构造的 IIR 滤波器保持线性相位，但不可能构造出在整个频带上保持线性相位的 IIR 滤波器，而 FIR 滤波器可以在整个频带上保持线性相位。

5.2.2　FIR 滤波器线性相位条件

相位延迟可以看作滤波器归一化相位响应的负值。也就是说，它是给定频率的相位响应除以该频率。滤波器的群延迟是相位曲线的负斜率。如果相位延迟是常量，群延迟必然是一个常量。然而，当相位延迟不是常量时，群延迟则可以是常量。某些系统要求其中一个或者两者都是常量。

如果数字滤波器拥有恒定的相位延迟和恒定的群延迟，那么其相位响应应该具有如下形式：

$$\theta(\Omega) = -m\Omega$$

式中，

$$\Omega = \omega T \tag{5.6}$$

该式表示的是一条斜率为 $-m$ 的直线，具有线性相位响应。

为了确定在什么条件下将会得到恒定相位延迟、恒定群延迟、线性相位响应，根据 FIR 滤波器的传输函数可得到其对应的频率响应。

$$H(z) = \sum_{n=0}^{L-1} h[n]z^{-n}$$

$$H(z)\big|_{z=e^{j\Omega}} = \sum_{n=0}^{L-1} h[n]e^{-j\Omega n} = \sum_{n=0}^{L-1} h[n][\cos(\Omega n) - j\sin(\Omega n)]$$

根据 FIR 滤波器的频率响应，可以得到其相位响应如下：

$$\theta(\Omega) = \angle[H(e^{j\Omega})] = \arctan\left[\frac{-\sum_{n=0}^{L-1} h[n]\sin(\Omega n)}{\sum_{n=0}^{L-1} h[n]\cos(\Omega n)}\right] = -m\Omega \tag{5.7}$$

由于正切是正弦除以余弦，于是可以改写成

$$\tan(\theta(\Omega)) = \frac{-\sum_{n=0}^{L-1} h[n]\sin(\Omega n)}{\sum_{n=0}^{L-1} h[n]\cos(\Omega n)} = \frac{\sin(-m\Omega)}{\cos(-m\Omega)}$$

通过交叉相乘后，将所有项都移到等式的左边，得到两个角度之和形式的正弦函数，

$$\sum_{n=0}^{L-1} h[n]\sin(m\Omega)\cos(\Omega n) - \sum_{n=0}^{L-1} h[n]\sin(\Omega n)\cos(m\Omega) = 0$$

应用三角函数积化和差关系，可得

$$\sum_{n=0}^{L-1} h[n]\sin(m\Omega - n\Omega) = 0$$

展开此方程，方程的解便很显然。

$$\begin{aligned} &h[0]\sin(m\Omega) + h[1]\sin(m\Omega - \Omega) + h[2]\sin(m\Omega - 2\Omega) + \cdots \\ &+ h[L-2]\sin(m\Omega - \Omega[L-2]) + h[L-1]\sin(m\Omega - \Omega[L-1]) = 0 \end{aligned}$$

可以发现，如下条件可满足该方程：

$$\sin(m\Omega) = -\sin(m\Omega - \Omega[L-1]) \text{ 和 } h[0] = h[L-1]$$

$$\sin(m\Omega - \Omega) = -\sin(m\Omega - \Omega[L-2]) \text{ 和 } h[1] = h[L-2]$$

$$\cdots$$

$$\sin(m\Omega - n\Omega) = -\sin(m\Omega - \Omega[L-n-1]) \text{ 和 } h[n] = h[L-n-1]$$

或

$$m\Omega - n\Omega = -m\Omega + \Omega(L-n-1)$$

$$m = \frac{(L-1)}{2} \text{ 和 } h[n] = h[L-n-1] \tag{5.8}$$

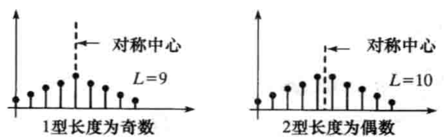
式(5.8)给出了 FIR 滤波器具有线性相位的条件：此式规定，脉冲响应 $h[n]$必须是对称的。有两种方法实现对称的脉冲响应，取决于 L 是偶数还是奇数。图 5.5 给出了奇数和偶数长度分别对应的脉冲响应对称的形式。

长度为奇数(偶数阶)的 FIR 滤波器的脉冲响应是对称的，如图 5.5a 所示，称为 1 型滤波器。如果长度是偶数(奇数阶)，则称为 2 型滤波器。例 5-2 为 1 型滤波器。

在决定线性相位的必要条件时，假定相位响应的形式是 $\theta(\Omega) = -m\Omega$。它是过原点的一条直线方程，具有恒定的相位延迟、恒定的群延迟以及线性相位响应。当然，可以使用形如$\theta(\Omega) = -m\Omega + \theta_0$的直线方程，其中 θ_0 是常量。如果使用该方程，则相位延迟不再恒定，而是关于 ω 的函数。群延迟仍然是常量。如果式(5.7)使用该条件，可以得到如下的结果(取 $\theta_0 = \pm\frac{\pi}{2}$)：

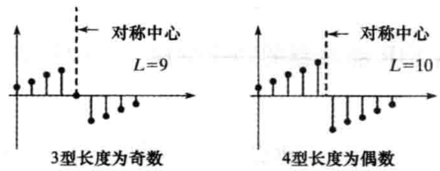
$$m = \frac{(L-1)}{2} \text{ 和 } h[n] = -h[L-n-1] \tag{5.9}$$

该脉冲响应称为反对称的(即奇对称)。典型的脉冲响应函数如图 5.6 所示，其中，长度为奇数时称为 3 型，长度为偶数时称为 4 型。



a）奇数长度　b）偶数长度

图 5.5　典型的脉冲响应序列，两种序列都是对称的且都能产生线性相位



a）奇数长度　b）偶数长度

图 5.6　典型的脉冲响应序列，两种序列都是反对称的且都能产生线性相位

1 型与 2 型滤波器的频率响应可以写成余弦函数和的形式，而 3 型和 4 型滤波器的频率响应则可写成正弦函数和的形式。4 种滤波器的频率响应在其应用时存在一些限制，如表 5.2 所示。

表 5.2　几种线性相位 FIR 滤波器的限制与适用条件

类型	对称性	限制条件	适用情况
1 型	奇数长度对称	无	低通 高通 带通 带阻
2 型	偶数长度对称	$H(f_s/2)=0$	低通 带通
3 型	奇数长度反对称	$H(0)=0$ $H(f_s/2)=0$	高通 微分器
4 型	偶数长度反对称	$H(0)=0$	高通 微分器

对于 2 型和 4 型 FIR 滤波器，其长度为偶数，需要对滤波器的脉冲响应在时间上进行平移抽样以得到因果的滤波器。例如图 5.6b 中，对称中心介于两个样本之间，因此滤波器必须平移 4 个半样本以满足因果关系。偶数长度的 FIR 滤波器可由一个奇数长度滤波器

的脉冲响应经重新抽样生成，通过该方式可以实现偶对称，过程如例 5-3 所示。

例 5-3 线性相位 FIR 滤波器

选择抽样频率 $f_s=11025\text{Hz}$，截止频率 $f_c=2000\text{Hz}$，阶数 $N=9$(长度=10)，使用傅里叶级数方法求解 z 域传输函数的系数，理想滤波器特征如图 5.2 所示。

设计 2 型线性相位低通 FIR 滤波器，长度为 10，抽样率为 11025Hz，截止频率为 2000Hz。

解：该滤波器设计指标除了长度是偶数而不是奇数以外，其他与例 5-2 相同。因此可以使用相同的过程求解脉冲响应。

对于脉冲响应进行重新抽样，利用每隔一个样本方式获得偶对称。因为使用每隔一个样本的方式，所以抽样率应变为原抽样率的一半。因此对应于此题，原先的抽样率应该为 22050Hz。脉冲响应如下：

$$C_k=\frac{1}{f_s}\int_{-f_c}^{f_c}(1)e^{-jk2\pi f/f_s}df=\frac{1}{k\pi}\sin(2\pi kf_c/f_s)$$

设置 $f_s=22050$，$f_c=2000$，可以计算得到脉冲响应如表 5.3 所示。

表 5.3 抽样频率为 22050Hz 的滤波器的脉冲响应幅值

	i	C_i		i	C_i
	0	0.18140	→	5，−5	0.01833
→	1，−1	0.17174		6，−6	−0.01455
	2，−2	0.14460	→	7，−7	−0.03409
→	3，−3	0.10508		8，−8	−0.03932
	4，−4	0.06040	→	9，−9	−0.03234

箭头指向的重抽样点构造的滤波器是对称的且长度为偶数。

图 5.7 所示的是如何选择样本点以得到长度为偶数的滤波器。设置抽样率为 11025Hz，对脉冲响应进行重抽样，重抽样点如表 5.3 中箭头和图 5.7 所示，这样选择的样本点构造了一个对称的脉冲响应。图 5.8 给出了求解结果对应的幅度谱与相位谱。

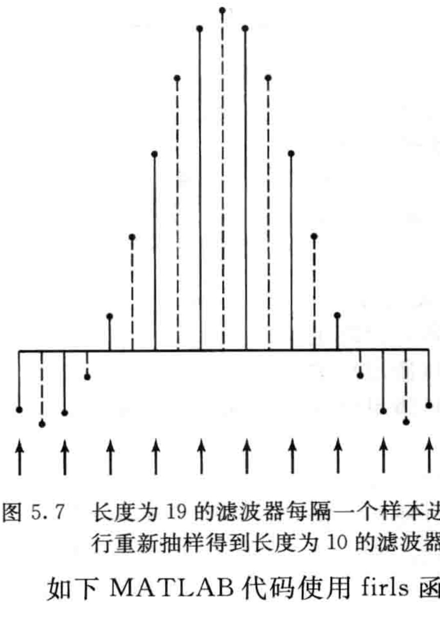


图 5.7 长度为 19 的滤波器每隔一个样本进行重新抽样得到长度为 10 的滤波器

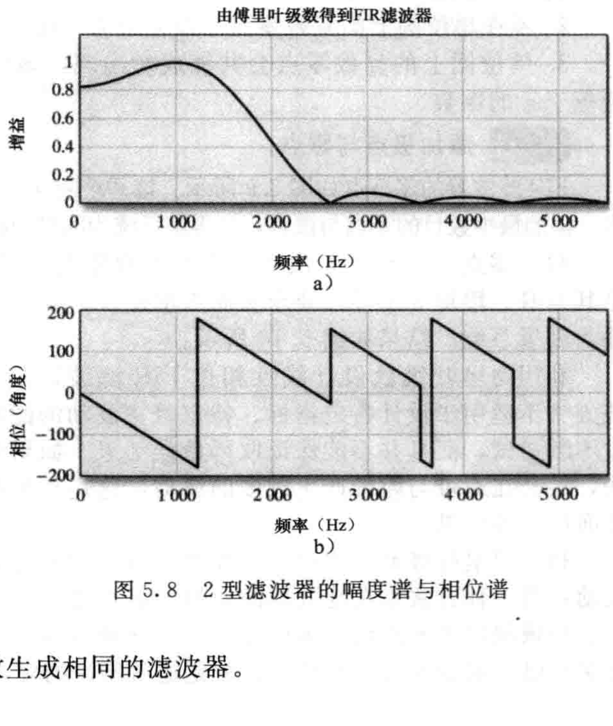


图 5.8 2 型滤波器的幅度谱与相位谱

如下 MATLAB 代码使用 firls 函数生成相同的滤波器。

```
N = 9;                                  %Order
Fs = 11025;
F = [0 2000/(fs/2) 2000/(fs/2) 1]; %Frequency vector normalized to fs/2
M = [1 1 0 0];                          %Corresspondîng magnitude vector
num = firls(N, F, M);
[H f] = freqz(num, 1, 1024, fs);
plot(f, abs(H));
```

◀

5.2.3 对称约束

为了说明脉冲响应的对称性对传输函数零点位置的约束，下面举例讨论一个长度为5且对称的传输函数。

$$H(z)=K\frac{h[0]z^4+h[1]z^3+h[2]z^2+h[3]z+h[4]}{z^4}=K\frac{N(z)}{D(z)}$$

如果滤波器的脉冲响应是对称的，即 $h[0]=h[4]$，$h[1]=h[3]$，利用该对称条件，并对分子多项式提取 z^2，得到下式

$$N(z)=z^2[h[0]z^2+h[1]z+h[2]+h[1]z^{-1}+h[0]z^{-2}]$$

使用 z 的倒数替换 z

$$N(z^{-1})=z^{-2}[h[0]z^{-2}+h[1]z^{-1}+h[2]+h[1]z+h[0]z^2]$$

推广至长度为 L 的滤波器为：

$$N(z)=z^{-m}[h[m]+\sum_{n=0}^{m-1}h[n](z^{-(m-n)}+z^{(m-n)})],\text{其中 } m=\frac{L-1}{2}$$

当 L 是偶数或者反对称时可得到相似的结果。

由于使用 z 的倒数代替 z 产生了相同的结果，所以可以发现，如果有一个零点在 z_1 位置，那么在 $1/z_1$ 位置同样会出现一个零点。

要求传输函数为实系数，并结合零点位置的约束，可以得到如下关于零点位置的结论，线性相位FIR数字滤波器的零点位置：

1. 实数的零点伴随着它的倒数同时出现，表明除了当 $z=\pm1$（它们的倒数是其自己）时，它们总是成对出现。

2. 不在单位圆上的复数零点总以4个为一组出现。

3. 单位圆上的复数零点会共轭成对出现，该情况下 z 的共轭等于 z 的倒数。

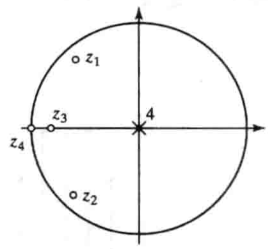


图5.9　增加最少数量的零极点使得系统因果且具有线性相位

例5-4　添加极点与零点

FIR滤波器的零极点如图5.9所示。该系统有4个零点和4个极点，添加最少数目的零点与极点，使得该系统为因果的线性相位系统。

解： 零点 z_1、z_2、z_3 没有对应的倒数零点，零点 z_4 的倒数是其本身。增加3个零点要求在原点增加3个极点，以使得滤波器为因果系统。结果如图5.10所示。◀

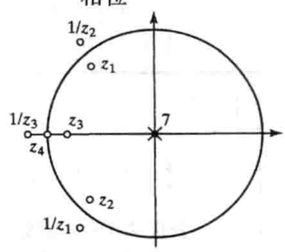


图5.10　增加3个极点与3个零点

利用傅里叶级数设计线性相位FIR滤波器是直接的方法，该方法不适用于设计特定指标、特定带宽波动的滤波器，只能通过不断尝试，而且并不能保证取得最佳结果。如果单独使用该方法，还会在通带与阻带产生过多的波动，这是直接截断傅里叶级数而产生的结果。

利用傅里叶级数方法设计滤波器，有两种方法可减少或消除波动：第一种方法是在滤波器的通带与阻带之间引入过渡带，由于在频域消除了不连续，因而设计的滤波器减少了波动；第二种方法提供了更令人满意的结果，就是通常所说的窗函数法。

5.2.4　窗函数与 FIR 滤波器

当利用傅里叶级数方法由理想滤波器生成 FIR 滤波器时，得到了从负无穷到正无穷的一系列系数(脉冲响应)。由于无法实现一个无限长度的滤波器。数学上这种模型是使用矩形窗乘以脉冲响应来近似实现的，如图 5.11 所示。这种时域的相乘在有限的带宽内保留了脉冲响应所有取值，并将其他项全部置零。为了更好地理解此过程，可以在频域观察对脉冲响应矩形加窗的影响，此时时域的乘法对应于频域的卷积。

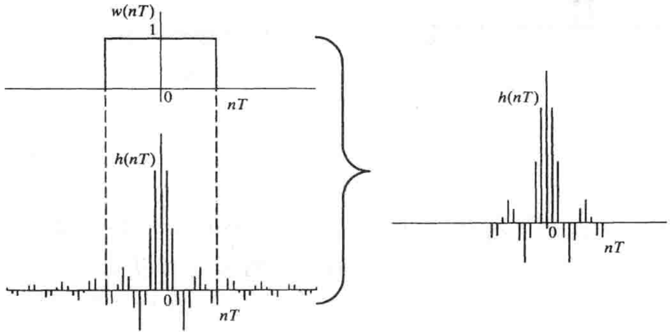


图 5.11　窗函数(上)乘以脉冲响应函数(下)产生有限项脉冲响应

时域矩形窗的离散傅里叶变换(DFT)在频域是抽样函数，图 5.12 说明了整个过程。在图 5.12中利用图形卷积说明处理过程，在频域将矩形窗与理想滤波器卷积，两个离散信号的卷积公式为：

$$Y[n]=H*W=\sum_{k=0}^{\infty}H[k]\cdot W[n-k]$$

其中，$H(k)$是理想滤波器的频域形式，$W(k)$是矩形窗的频域形式。函数 $W(-k)$实现了窗函数的左右翻转，$W(n-k)$实现翻转函数右移 n 个单位。对于 $Y(n)$中每一个取值 n，都必须左右翻转$W(k)$后平移 n 个单位，再乘以 $H(k)$，最后再将结果求和。

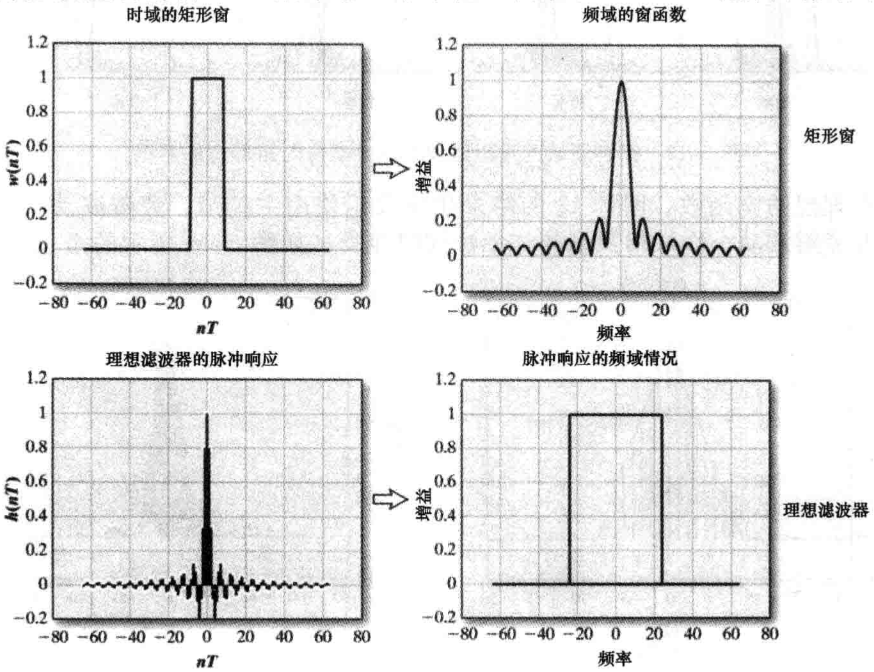


图 5.12　矩形窗作用于理想低通滤波器

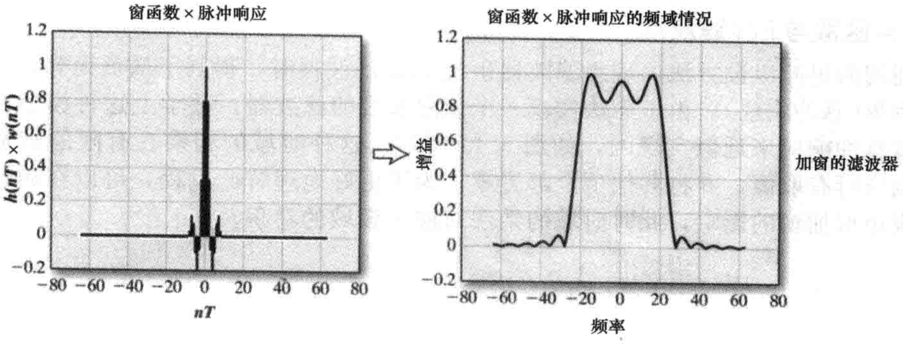


图 5.12 (续)

图 5.12 中左列是时域，右列是对应的频域。加窗的滤波器在频域是理想滤波器与矩形窗的卷积。注意图 5.12 中的波动和倾斜过渡带。

图 5.13 以图形方式描述该过程，其中阴影部分表明对于矩形窗函数上的点乘以相应理想滤波器上的点，然后对每个 n 值进行求和。从图形上可以明显发现，最终滤波器的波动是由窗函数的波动所引起，滤波器过渡带的宽度与窗函数主瓣宽度有关。在理想情况下，希望窗函数在频域上是冲激信号，原因在于冲激信号与理想滤波器的卷积结果仍为理想滤波器。但频域内冲激窗函数对应时域信号为无限长。

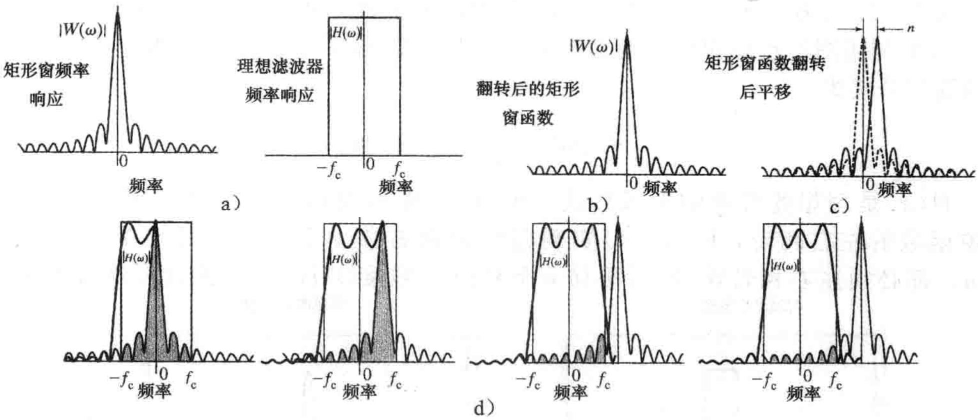


图 5.13　图解频域内理想滤波器脉冲响应与矩形窗的卷积

不存在理想的窗函数，图 5.13 的结果建议我们使用主瓣窄、旁瓣波动小的窗函数。但主瓣窄与旁瓣波动小恰好相互制约，无法同时兼得。如图 5.14 所示的是三角窗，三角

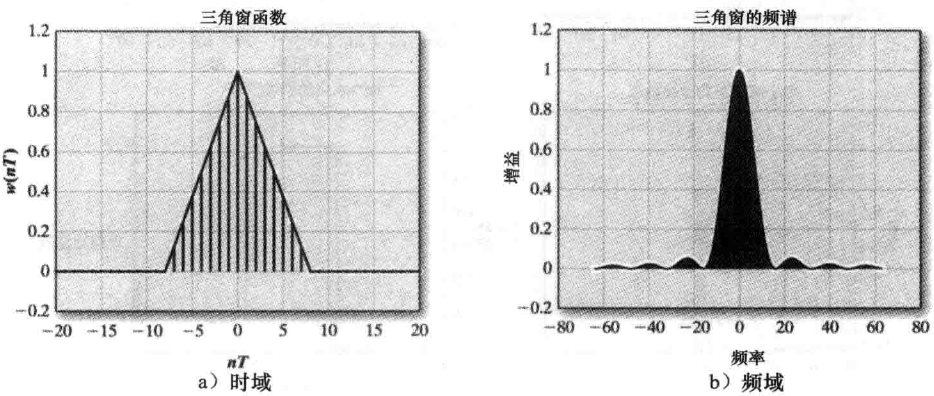


图 5.14　三角窗函数

窗将窗口外脉冲响应置为零从而截断脉冲响应，三角窗函数的信号幅值成线性衰减。在频域我们可以看到，三角窗相比于矩形窗，其旁瓣波动减小但其主瓣变得更宽。使用三角窗函数与理想低通滤波器来设计 FIR 数字滤波器，该滤波器将具有较小的波动，但将会产生更宽的过渡带，如图 5.15 所示。

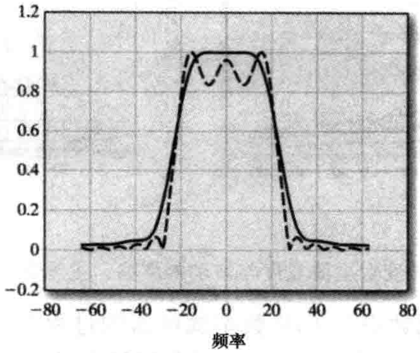


图 5.15 利用三角窗与理想滤波器构造滤波器，虚线是矩形窗构造的滤波器

除了矩形窗和三角窗，还有其他的窗函数可供使用，它们为滤波器设计提供了在波动与过渡带之间的多种平衡。表 5.4 给出了 5 种常用的窗函数及其时域表示。

表 5.4 常用窗函数

窗函数名称	函数	MATLAB 中的调用语句
布莱克曼窗	$w_B[n]=\begin{cases}0.42+0.5\cos\left(\frac{2\pi n}{L-1}\right)+0.8\cos\left(\frac{4\pi n}{L-1}\right), & \lvert n\rvert\leqslant(L-1)/2\\0, & \text{其他}\end{cases}$	[w] = blackman(L);
汉明窗和汉宁窗	$w_H[n]=\begin{cases}\alpha+(1-\alpha)\cos\left(\frac{2\pi n}{L-1}\right), & \lvert n\rvert\leqslant(L-1)/2\\0, & \text{其他}\end{cases}$ 其中，对于汉明窗 $\alpha=0.54$，对于汉宁窗 $\alpha=0.5$.	w = hamming(L); w = hann(L);
矩形窗	$w_R[n]=\begin{cases}1, & \lvert n\rvert\leqslant(L-1)/2\\0, & \text{其他}\end{cases}$	w = boxcar(L);
三角窗①	$w_T[n]=\begin{cases}\frac{-2\lvert n\rvert}{L+1}+1, & \lvert n\rvert\leqslant(L-1)/2\\0, & \text{其他}\end{cases}$	w = triang(L);

①三角窗函数有另一个版本，叫做 Bartlett 窗，它有略微不同的端点。
w=bartlett(L);

对于表 5.4 所示的窗函数，布莱克曼窗拥有最宽的主瓣和最小的波动。矩形窗拥有最窄的主瓣和最大的波动。主瓣宽度和波动可由图 5.16 评定。主瓣宽度是第一个零点之间的距离。通常波动使用旁瓣中最大的波动评定。图 5.17 对比了 5 种窗函数的波动与主瓣宽度(长度从 11 到 99，仅考虑了奇数阶长度)。

图 5.17 中，瓣宽度与长度曲线可以由下式估算：

$$\omega\cdot l=K$$

其中，ω 是瓣宽度，l 是滤波器长度，K 是由窗类型确定的常量。表 5.5 随着波动比例给出了常量 K 的取值。

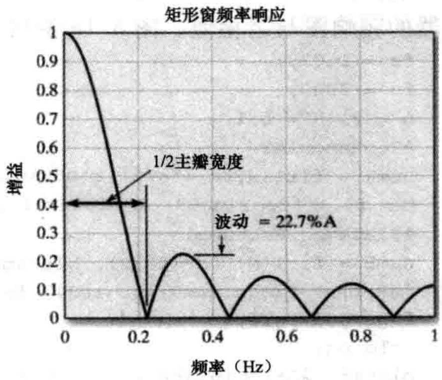


图 5.16 频域内长度为 9 的矩形窗。第一个零点的宽度看作主瓣宽度，波动由最大波动定义

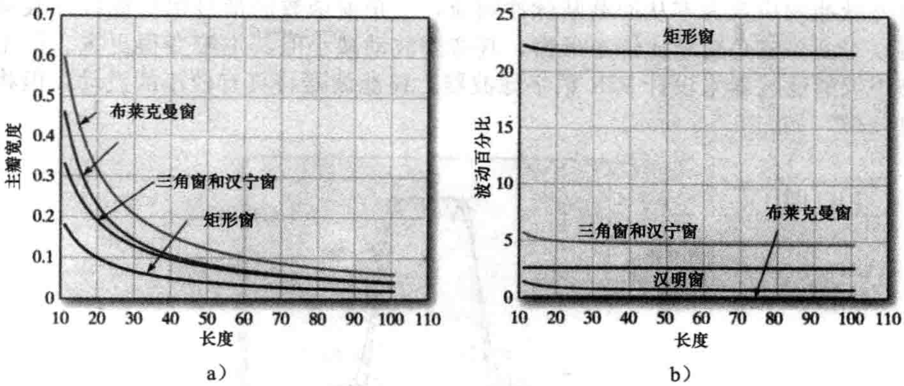


图 5.17　窗长度对主瓣宽度与波动的影响。主瓣宽度已归一化到 f_S

总体来说，在利用窗函数设计 FIR 数字滤波器的过程中，首先选择理想滤波器并确定脉冲响应函数(通常是无限长且非因果的)。再利用该脉冲响应在时域逐项乘以窗函数以得到有限长度的脉冲响应。平移产生的脉冲响应以得到因果的脉冲响应。该设计过程通过手工计算将非常乏味并且困难，整个过程可以由 MATLAB 中的 firl 函数实现。例 5-5 详细说明了这个函数。

表 5.5　不同类型窗 K 的取值，瓣宽度乘以滤波器长度近似为常量

窗类型	K	波动(%)	波动(dB)
矩形窗	2.0	21.75	13.2
三角窗	3.8	4.76	26.4
汉宁窗	3.9	2.67	31.5
汉明窗	4.4	0.77	42.3
布莱克曼窗	6.1	0.12	58.4

例 5-5　设计加窗的滤波器

理想低通滤波器的抽样频率为 11025Hz，截止频率为 2000Hz。根据理想滤波器利用以下窗函数设计相应的数字滤波器。矩形窗、三角窗、汉明窗、汉宁窗、布莱克曼窗，所有滤波器长度为 21。

解： 如下 MATLAB 代码显示了由矩形窗和布莱克曼窗设计的滤波器的频响，其他滤波器的频响图与之相似。图 5.18 给出了设计结果。

```
fs = 11025;
fc = 2000;
L = 21;N = L-1;
%Rectangular
numR = fir1(N,fc/(fs/2),rectwin(L));
[Hr f] = freqz(numR, 1, 1024, fs);
%Blackman/Rectangle
numB = fir1(N, fc/(fs/2), blackman(L));
[HB f] = freqz(numB, 1, 1024, fs);
figure(1);clf;
hold on;
plot(f, 20*log10(abs(HB)), 'k');
plot(f, 20*log10(abs(Hr)), 'k');
```

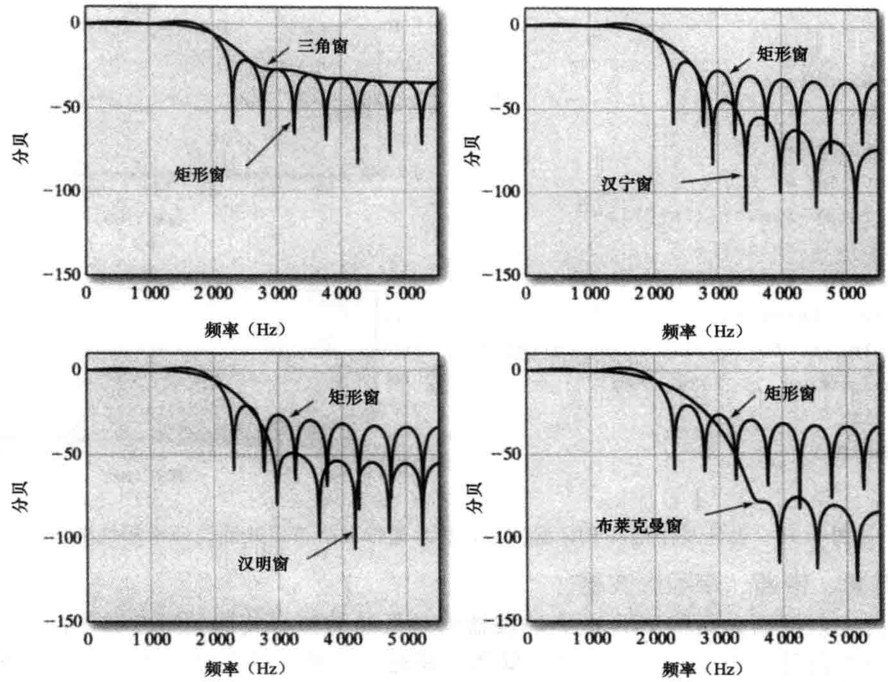


图 5.18 每个滤波器都是基于相同的理想低通滤波器，三角窗、汉明窗、汉宁窗、布莱克曼窗对比矩形窗的频率响应

窗函数法设计 FIR 数字滤波器的难点在于确定正确的阶数，一些尝试是必须的过程，表 5.5提供了一些指导，步骤总结以下：

1. 将阻带波动转换到分贝形式以符合表 5.5。

2. 从表 5.5 中选择窗函数，需要注意的是，表 5.5 中的值是近似的。表 5.5 提供了 K 的取值。

3. 确定 $\Delta f=f_{stop}-f_{pass}$ 的取值，Δf 必须归一化使得 $f_s/2=1$。

4. 使用 $K/\Delta f$ 确定滤波器长度。

5. 由通带和阻带边界频率的平均值计算截止频率。

6. 使用 MATLAB 中的 fir1 函数设计滤波器，生成通带与阻带的分解视图，以确定生成的滤波器是否满足设计指标，并相应地调整滤波器长度。

例 5-6 设计线性 FIR 滤波器

设计线性相位 FIR 滤波器满足以下指标：抽样频率 11025Hz，通带从 0Hz 到 1650Hz，阻带从 2000Hz 到 $f_s/2$，通带与阻带的波动为 0.01。

解： 归一化 Δf 的值 $(2000-1650)/(f_s/2)=0.06349$。阻带波动 0.01 对应于阻带 40dB 衰减。根据表 5.5 可知，矩形窗与三角窗不能满足要求。因为实际的阻带衰减通常会比最大波动衰减小 10dB 左右，所以汉宁窗可以满足需求。由表 5.5 中的汉明窗可知，滤波器长度是 $4.4/\Delta f=69.3$。MATLAB 的 fir1 函数设计滤波器阶数比长度少 1。所以第一次尝试，设置滤波器阶数为 69。fir1 函数还需要提供截止频率，使用 $f_c=(f_{pass}+f_{stop})/2=1825$。MATLAB 函数设计的滤波器如下：

```
num= fir1(69,1825,hamming(70));
```

使用该函数产生的滤波器的波动满足要求，但是过渡带过大。增加滤波器阶数为 88 生成的滤波器满足所有指标，图 5.19 给出了结果。

```
%FIRDesignExample.m
fs = 11025;
fpass = 1650;Rp = .01;
fstop = 2000;Rs = .01;
deltaf =((fstop-fpass)/(fs/2));
len = round(4.4/deltaf);
len = len + 20;
fc = (fpass + fstop)/2;
num = fir1(len-1, fc/(fs/2), hamming(len));
[H f] = freqz(num, 1, 1024, fs);
figure(1);clf;
plot(f, abs(H));
```

a)

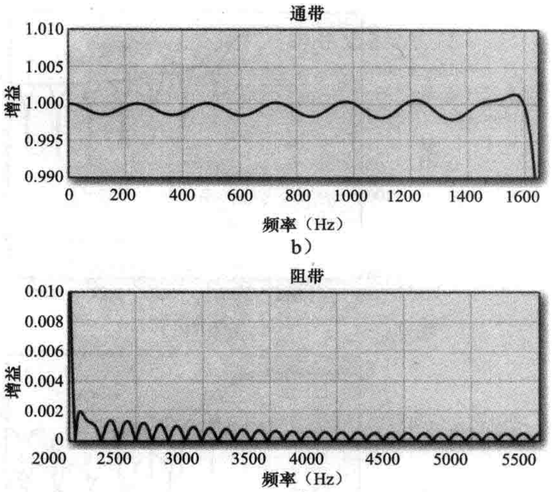
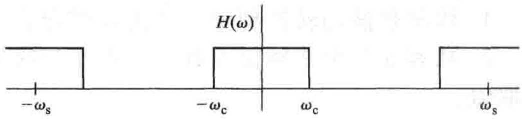
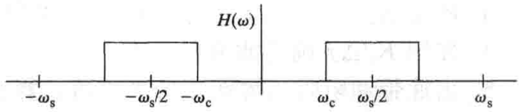
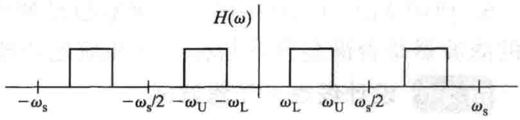
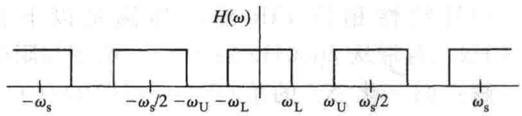


图 5.19 设计线性相位 FIR 滤波器，显示通带和阻带以验证设计指标均满足

5.2.5 高通、带通、带阻滤波器

到此为止，前面介绍的 FIR 数字滤波器均是低通类型，其他类型的数字滤波器可以使用相同的方法设计。表 5.6 概述了对于低通、高通、带通、带阻滤波器，使用傅里叶级数方法所得到的脉冲响应，MATLAB 中的 fir1 函数可以由函数选项实现这些公式。例 5-7 说明了设计带通滤波器的过程。

表 5.6 利用傅里叶级数设计的低通、高通、带通和带阻滤波器的脉冲响应。一旦脉冲响应确定，应用相应的窗函数即可获得所需的滤波器

滤波器	$H(\omega)$
低通 $h[k]=\dfrac{\sin(2\pi k f_c/f_s)}{k\pi}$	
高通 $h[k]=\delta[k]-\dfrac{\sin(2\pi k f_c/f_s)}{k\pi}$	
带通 $h[k]=\dfrac{\sin(2\pi k f_U/f_2)-\sin(2\pi k f_L/f_s)}{k\pi}$	
带阻 $h[k]=\delta[k]-\dfrac{\sin(2\pi k f_U/f_2)-\sin(2\pi k f_L/f_s)}{k\pi}$	

例 5-7 利用汉明窗设计带通滤波器

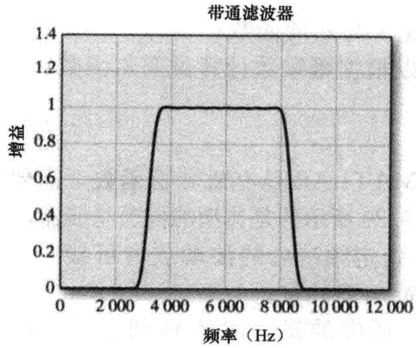
使用傅里叶级数方法设计满足下列指标的带通滤波器

抽样频率	22050Hz
第一阻带	0～2800Hz
通带	3600～8000Hz
第二阻带	8800Hz～$f_s/2$
所有频带中的波动	0.01

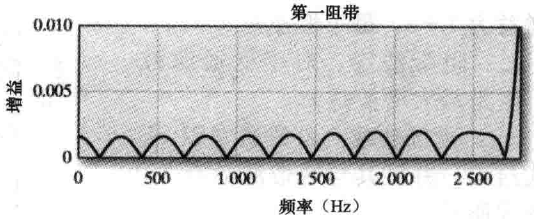
解：此题中的上、下过渡带的宽度相同(如果宽度不一致，可以使用最窄的带宽作为最差情况下的设计)。归一化的过渡带宽度是 $\Delta f=(3600-2800)/11025=0.0726$，对于汉明窗，滤波器的长度为 $4.4/\Delta f=60.6$。下面设计 61 阶的滤波器进行尝试，检查是否满足设计指标。

```
fs = 22050;
fst1 = 2800;
fps1 = 3600;fps2 = 8000;
fst2 = 8800;
Rp = .01;
deltaf =((fps1-fst1)/(fs/2));
len = round(4.4/deltaf);
len = len + 20;
disp(len);
fc1 = (fps1 + fst1)/2;
fc2 = (fps2 + fst2)/2;
fn = [fc1 fc2]/(fs/2);
num = fir1(len-1, , fn, ...
        'bandpass', hamming(len));
[H f] = freqz(num, 1, 1024, fs);
figure(1);clf;
plot(f, abs(H));
figure(2);clf;
subplot(3, 1, 1);
plot(f, abs(H));
axis([0 fst1 0 Rp]);
xlabel('frequency in Hz');
ylabel('gain');
title('First stop band');
subplot(3, 1, 2);
plot(f, abs(H));
axis([fps1 fps2 1-Rp 1+Rp]);
xlabel('frequency in Hz');
ylabel('gain');
title('Pass band');
subplot(3, 1, 3);
plot(f, abs(H));
axis([fst2 fs/2 0 Rp]);
xlabel('frequency in Hz');
ylabel('gain');
title('Second stop band');
```

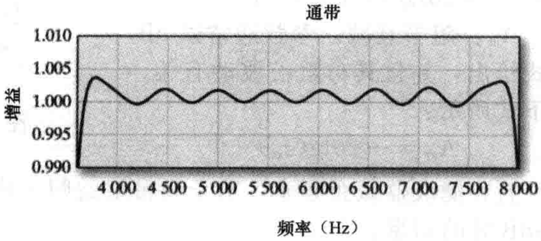
a)



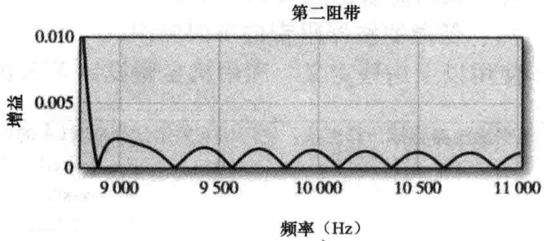
b)



c)



d)



e)

5.3 高级窗函数

5.3.1 凯瑟窗

到目前为止，还没有一种窗函数能够提供简便的机制用于提前指定滤波器的参数。因此，设计时往往要经过不断尝试和良好的直觉。凯瑟窗与之前的窗函数有所不同，通过改变它的参数可以实现给定的波动指标。凯瑟窗由贝塞尔函数构成，表示式如下：

$$w_k[n]=\frac{I_0(\beta)}{I_0(\alpha)}$$

其中，$I_0(\alpha)$与$I_0(\beta)$是零阶修正的第一类贝塞尔函数，常量α用来调整旁瓣的衰减而且与参数β有关，两者关系如下：

$$\beta=\alpha\sqrt{1-\left(\frac{2n}{L-1}\right)^2}$$

可见β是α与n的函数。

可以根据幂级数计算贝塞尔函数：

$$I_0(x)=\sum_{k=0}^{n}\left[\frac{(x/2)^k}{k!}\right]^2$$

此外，MATLAB具有凯瑟窗函数，需要提供滤波器长度与参量α作为变量。

图5.20所示的是典型的低通滤波器，并以符号和定义指定参数。根据此图给出如下定义。

ε：指定频带内的波动或者可接受的上冲与下冲量。

ε_p：通带波动。对于低通滤波器，通带增益归一化为1；通带最大增益为$1+\varepsilon$，最小增益是$1-\varepsilon$。

ε_{st}：阻带波动。对于低通滤波器，阻带最大增益是ε_{st}。

A_p：通带衰减。参数通常以dB形式给出，并且其与通带波动有关，如下式所示：

$$A_p=-20\lg[(1+\varepsilon_p)/(1-\varepsilon_p)]$$

A_{st}：阻带衰减。参数通常以dB形式给出，并且其与阻带波动有关，如下式所示：

$$A_{st}=-20\lg(\varepsilon_{st})$$

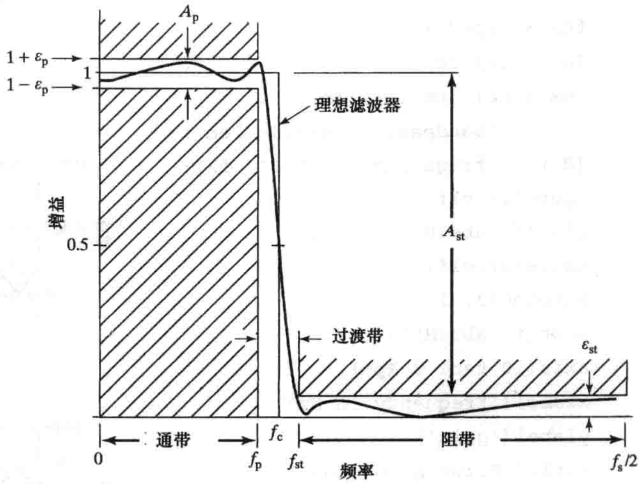


图5.20　利用凯瑟窗设计低通滤波器的变量

f_c：滤波器截止频率。对于通带增益归一化的低通滤波器来说，此频率对应于增益为0.5dB时的频率。

f_p：低通滤波器通带的上限频率。

f_{st}：低通滤波器阻带的下限频率。

使用以上这些定义，利用凯瑟窗设计FIR滤波器的流程如图5.21所示。

1. 根据滤波器指标计算衰减，使用如下关系计算衰减(A的单位是分贝)；

$$A=\max(A_p,A_{st})\text{ 或者，}$$
$$\varepsilon=\max(\varepsilon_p,\varepsilon_{st})$$
$$A=-20\lg(\varepsilon)$$

2. 根据如下公式计算α；

$$\alpha=\begin{cases}0.1102(A-8.7) & A\geqslant 50\\ 0.5482(A-21)^4+0.7886(A-21) & 21<A<50\\ 0 & A\leqslant 21\end{cases}$$

3. 根据如下公式计算滤波器长度L：L的值应该向上取整为临近的奇整数；

$$L=1+\frac{K_A f_s}{f_{st}-f_p},\text{其中},K_A=\begin{cases}\dfrac{A-7.95}{14.36}, & A>21\\ 0.922, & \text{其他}\end{cases}$$

图5.21　利用凯瑟窗设计FIR滤波器的流程

4. 根据如下关系式计算凯瑟窗函数，也可以使用 MATLAB 命令来计算凯瑟窗函数：

$$w_K[n]=\frac{I_0(\beta)}{I_0(\alpha)}$$

$$\beta=\alpha\sqrt{1-\left(\frac{2n}{L-1}\right)^2}$$

$$I_0(x)=\sum_{k=0}^{\infty}\left[\frac{(x/2)^k}{k!}\right]^2$$

5. 通过傅里叶级数计算满足理想滤波器设计指标的脉冲响应函数；对于长度为奇数的理想低通滤波器，滤波器的系数可有如下公式计算：

$$C_k=\frac{1}{k\pi}\sin(2\pi kf_c/f_s)$$

6. 将相应的窗函数系数乘以脉冲响应系数，从而得到最终的因果有限长滤波器的系数。可以使用 MATLAB 中的 fir1 函数构造凯瑟窗滤波器

图 5.21 （续）

例 5-8 利用凯瑟窗设计低通滤波器

利用凯瑟窗设计低通 FIR 数字滤波器，使得其满足如下指标：

抽样频率：$f_s=11025$Hz

通带：0～2500Hz　　通带波动：±0.01

阻带：3400Hz～$f_s/2$　　阻带波动：0.03

图 5.21 显示了设计流程。

解：如下 MATLAB 代码使用 fir1 函数与凯瑟窗函数设计 FIR 滤波器，设计结果的频率响应如图 5.22 所示。

```
fs = 11025;
fpass = 2500;
fstop = 3400;
fc = fpass + (fstop-fpass)/2;
rp = .01;rs = .03;
eps = min(rp, rs);
A = -20*log10(eps);
Ka = (A - 7.95)/14.36;
alpha = .5842*(A - 21)^.4 + .07886*(A - 21);
L = ceil(1 + Ka*fs/(fstop - fpass));
num = fir1(L-1, fc/(fs/2), kaiser(L, alpha));
[Hk f] = freqz(num, 1, 1024, fs);
plot(f, abs(Hk));
```

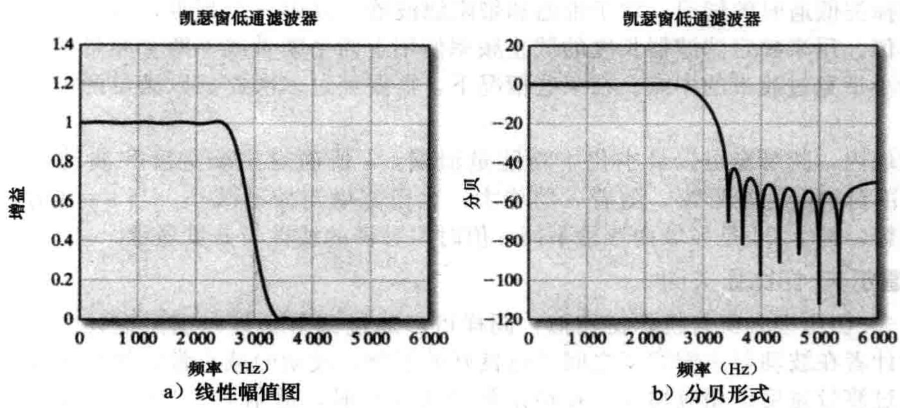
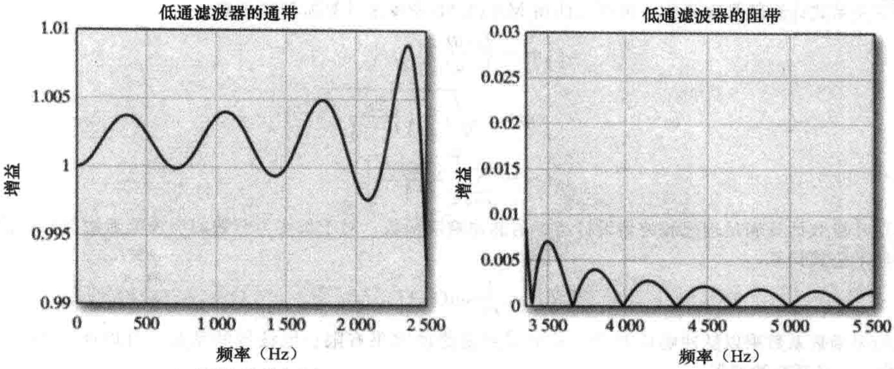


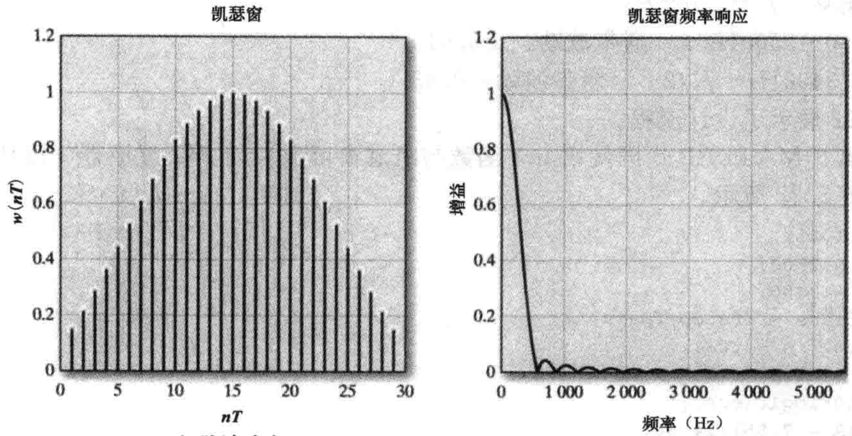
图 5.22　利用凯瑟窗设计的 29 阶 FIR 滤波器



c）通带的分解显示　　d）阻带的分解显示

图 5.22 （续）

凯瑟窗的形状由待设计的滤波器的参数所决定。此例所设计出的滤波器的频域与时域响应如图 5.23 所示。



a）脉冲响应　　b）频率响应

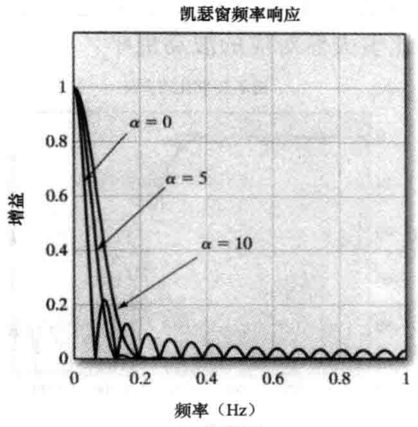
图 5.23 凯瑟窗

凯瑟窗同样可以适用于高通、带通和带阻滤波器。对于高通滤波器，滤波器长度与参数 α 的选择按低通时的情况。对于带通和带阻滤波器，其由三个频带，波动选择 3 个频带中的最小值。用来确定滤波器长度的截止频率使用上两个频带或下两个频带的中点，通常也使用最小带宽过渡带的中点。在某些情况下，需要经过多次尝试以满足两个过渡带的带宽要求。

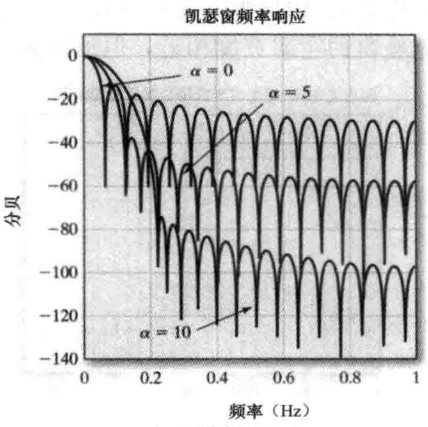
在频域内，凯瑟窗可以最小化主瓣能量泄漏。α 值确定主瓣宽度和波动百分比。当 $\alpha=0$时，凯瑟窗变为矩形窗。随着 α 的增大，主瓣宽度与波动减小。当 $\alpha=8$ 时，近似为布莱克曼窗。图 5.24 是频域内三种不同 α 值时凯瑟窗的线性与分贝显示。

5.3.2 道尔夫-切比雪夫窗

道尔夫-切比雪夫窗与凯瑟窗相似，同样可以通过调节参数以满足设计需求。窗函数法需要设计者在波动与主瓣宽度之间进行良好的平衡，波动的减小常以增加主瓣宽度为代价，同时过渡带宽度也相应增大。在给定的最大波动时，道尔夫-切比雪夫窗可以提供最窄的主瓣。



a）线性图



b）阻带波动的分贝显示

图 5.24 对于三种不同 α 值凯瑟窗的频率响应

道尔夫-切比雪夫窗的定义如下：

$$w_{DC}[n]=\frac{1}{L}\left[(1/\gamma)+2\sum_{k=1}^{(L-1)/2}V_N(\beta\cos(k\pi/L)\cos(2nk\pi/L)\right]$$

式中，

$$\beta=\cosh[(1/N)\operatorname{arcosh}(1/\gamma)]$$

$$\gamma=\frac{\text{旁瓣最大幅值}}{\text{主瓣最大幅值}}(\text{为最大波动百分比})$$

$$V_{N(x)}=\begin{cases}\cos(N\arccos(x)), & |x|\leqslant 1\\ \cosh(N\operatorname{arcosh}(x)), & |x|>1\end{cases}\ \text{定义一个切比雪夫多项式}$$

例 5-9 利用道尔夫-切比雪夫窗设计低通 FIR 滤波器

利用道尔夫-切比雪夫窗设计一个低通 FIR 滤波器，使其满足如下指标：

抽样频率：$f_s=11025\text{Hz}$

通带频率：0～2500Hz

通带波动：±0.01

阻带频率：3400Hz～$f_s/2$

阻带波动：0.03

解： 如下 MATLAB 代码使用 fir1 函数与道尔夫-切比雪夫窗函数设计 FIR 滤波器，计算结果的频率响应如图 5.25 所示。

```
fs = 11025;
fpass = 2500;
fstop = 3400;
fc = (fpass + fstop)/2;
rp = .01;rs = .03;
eps = min(rp, rs);
lambda = -20*log10(eps);
L = 34;
num = fir1(L-1, fc/(fs/2), chebwin(L, lambda));
[Hdc f] = freqz(num, 1, 1024, fs);
plot(f, abs(Hdc));
```

对于给定的波动百分比，道尔夫-切比雪夫窗在频域内拥有最小的主瓣宽度。参数 γ 的值确定波动的百分比。图 5.26 对比了长度为 31 的道尔夫-切比雪夫窗、矩形窗和布莱克曼窗。对于 γ=0.1，矩形窗与道尔夫-切比雪夫窗具有相似的主瓣宽度，但从图 5.26 中

可以看到，道尔夫-切比雪夫窗具有较小的最大波动。当$\gamma=0.0001$时，道尔夫-切比雪夫窗与布莱克曼窗的主瓣宽度相近，但道尔夫-切比雪夫窗对应的波动更小。

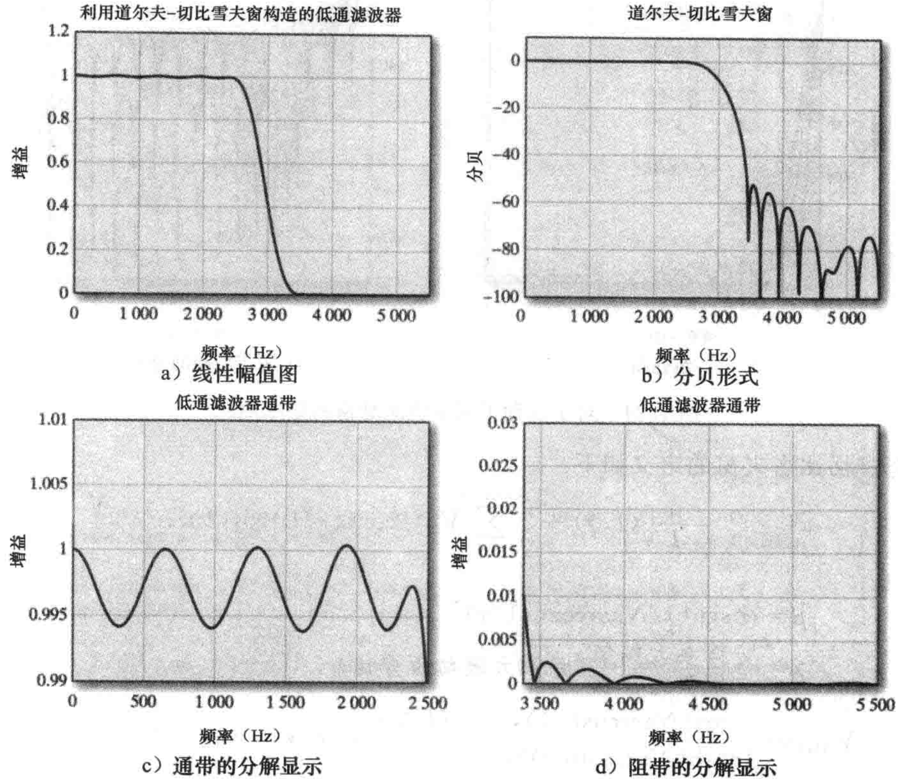


图 5.25　利用道尔夫-切比雪夫窗设计的 33 阶 FIR 滤波器

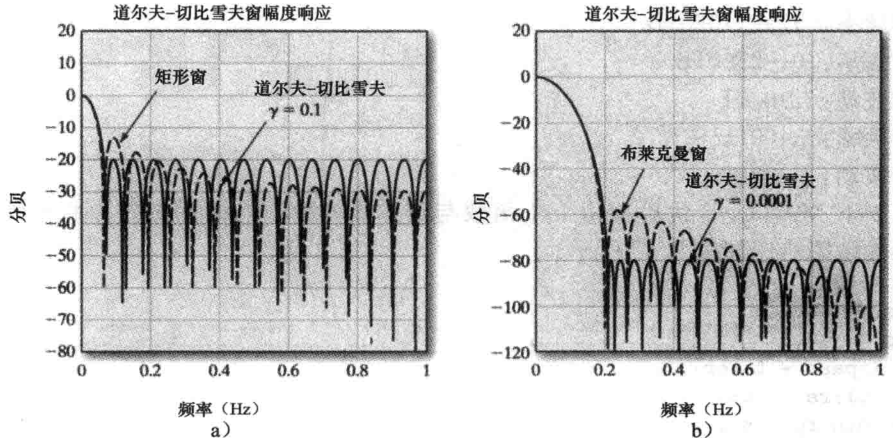


图 5.26　较矩形窗和布莱克曼窗与道尔夫-切比雪夫窗，它们具有相似的主瓣宽度。但道尔夫-切比雪夫窗得到了更低的最大波动百分比

5.4　频率抽样 FIR 滤波器

频率抽样是 FIR 数字滤波器的一种设计技术，其与傅里叶级数方法相关，其优点在于有时可以获得更好的结果并且适合于计算机优化。

频率采样法依赖于DFT，其以固定周期抽样滤波器理想的频率响应。通过样本的IDFT变换得到滤波器的脉冲响应，进而得到传输函数。因为希望得到实系数且线性相位的数字滤波器，所以需要一些约束条件。

IDFT的计算公式如下：

$$h[n]=\frac{1}{L}\sum_{k=0}^{L-1}H[k]e^{j\frac{2\pi kn}{L}}$$

式中，L是滤波器长度。

z变换公式为

$$H(z)=\sum_{n=0}^{L-1}h[n]z^{-n}$$

如果给定频率样本点$H[k]$的值，便可通过IDFT的公式计算得到$h[n]$，从而通过z变换求解滤波器的传输函数。因此

$$H(z)=\sum_{n=0}^{L-1}\left[\frac{1}{L}\sum_{k=0}^{L-1}H[k]e^{j\frac{2\pi kn}{L}}\right]z^{-n}$$

新排列求和顺序有

$$H(z)=\sum_{k=0}^{L-1}\frac{H[k]}{L}\sum_{n=0}^{L-1}e^{j\frac{2\pi kn}{L}}z^{-n}=\sum_{k=0}^{L-1}\frac{H[k]}{L}\sum_{n=0}^{L-1}W_k^n z^{-n}$$

式中，$W_k=e^{j\frac{2\pi k}{L}}$。该式经简化，有

$$H(z)=\sum_{k=0}^{L-1}\frac{H[k]}{L}\frac{z^L-1}{z^{L-1}(z-W_k)} \tag{5.10}$$

由于其具有除了原点之外的极点，因此该传输函数似乎是一个递归滤波器。需要注意的是，分子项z^L-1可以因式分解，它对应的L个根是$e^{j\frac{2\pi k}{L}}=W_k$。传输函数中的零点抵消了除原点之外的所有极点。展开$H(z)$的求和形式，进一步化简可得到如下非递归形式的传输函数：

$$H(z)=\frac{1}{Lz^{L-1}}\sum_{k=0}^{L-1}H[k]\left[\prod_{\substack{i=0\\i\neq k}}^{L-1}(z-e^{j\frac{2\pi i}{L}})\right] \tag{5.11}$$

因此，频率抽样法可用来设计具有有限脉冲响应的滤波器，递归形式使用式(5.10)，非递归形式使用式(5.11)。在任何情况下，滤波器总是稳定的(假定在递归形式中完全抵消了零极点)。

频率抽样法设计FIR数字滤波器的过程总结如下：

1. 从理想滤波器的频率响应上等间隔地选择L个样本组成$H[k]$。
2. 利用$h[n]=\frac{1}{L}\sum_{k=0}^{L-1}H[k]e^{j\frac{2\pi kn}{L}}$求得滤波器的脉冲响应。
3. 利用式(5.11)求得FIR滤波器的传输函数。

若以任意方式选取频率样本点，则可能导致设计出的滤波器具有虚系数或非线性相位。根据前面章节的分析，FIR数字滤波器若具有线性相位，其脉冲响应必须具有对称特性。若要得到实系数的FIR数字滤波器，其脉冲响应必须等于它的复共轭。这两个条件可以写成如下形式：

$$h[n]=h^*[n]$$
$$h[n]=h[L-n-1]$$

脉冲响应等于自身共轭的约束条件意味着样本点要么是实数要么是共轭对形式。写成如下公式形式：

$$H[k]=A(k)e^{j\theta(k)}=H^*(L-k)=A(L-k)e^{-j\theta(L-k)}$$

(可以写出 $H[k]$ 的 IDFT，然后令其等于其共轭来进行证明。)

对称条件对于 $\theta(k)$ 有约束。若要理解究竟是什么约束，考虑如下公式(假设 L 是奇数)：

$$h[n]=h^*[L-n-1]$$
$$\mathrm{DF}^{-1}\{h[n]\}=\mathrm{DF}^{-1}\{h^*[L-n-1]\}$$

或者

$$\frac{1}{L}\sum_{k=0}^{L-1}A(k)\mathrm{e}^{\mathrm{j}\theta(k)}\mathrm{e}^{\mathrm{j}\frac{2\pi kn}{L}}=\frac{1}{L}\sum_{k=0}^{L-1}A(k)\mathrm{e}^{-\mathrm{j}\theta(k)}\mathrm{e}^{-\mathrm{j}\frac{2\pi k}{L}(L-n-1)}$$

由方程两端指数项相等，得到

$$\theta(k)+\frac{2\pi kn}{L}=-\theta(k)-\frac{2\pi k}{L}(L-n-1)$$

由该公式可以确定 $\theta(k)$

$$\theta(k)=-\frac{L-1}{L}\pi k \tag{5.12}$$

式(5.12)给出了 $\theta(k)$ 的约束条件，可使得滤波器传输函数具有线性相位与实系数。

为了分析 $\theta(k)$ 的约束条件对脉冲响应的影响，采用约束条件下的 $\theta(k)$ 进行 IDFT 变换，求解相应的脉冲响应。

$$h[n]=\frac{1}{L}\sum_{k=0}^{L-1}A(k)\mathrm{e}^{-\mathrm{j}\frac{L-1}{L}\pi k}\mathrm{e}^{\mathrm{j}\frac{2\pi kn}{L}}=\frac{1}{L}\sum_{k=0}^{L-1}A(k)\mathrm{e}^{\mathrm{j}\left(\frac{2n+1}{L}\pi k-\pi k\right)}$$

因为 $\mathrm{e}^{-\mathrm{j}\pi k}=(-1)^k$，所以 $h[n]$ 可以写成

$$h[n]=\frac{1}{L}\sum_{k=0}^{L-1}(-1)^kA(k)\mathrm{e}^{\mathrm{j}\frac{2n+1}{L}\pi k}$$

已知 $A(k)=A(L-k)$ 和 $\mathrm{e}^{\mathrm{j}\frac{2n+1}{L}\pi k}=-\mathrm{e}^{-\mathrm{j}\frac{2n+1}{L}\pi(L-k)}$，可进一步简化该公式。应用这两个条件以余弦函数形式给出脉冲响应

$$h[n]=\frac{1}{L}\left[A(0)+2\sum_{k=1}^{\frac{L-1}{2}}(-1)^kA(k)\cos\frac{(2n+1)\pi k}{L}\right] \tag{5.13}$$

对于偶数长度的滤波器，其脉冲响应有相似的形式，只是存在些许不同。以相似的方式可考虑脉冲响应(奇数长度或者偶数长度)是反对称的情况。对以上 4 种情况的总结见表 5.7。

表 5.7 频率抽样法设计滤波器的脉冲响应

对称性	L	脉冲响应	约束
对称	奇数	$h[n]=\frac{1}{L}\left[A(0)+2\sum_{k=1}^{\frac{L-1}{2}}(-1)^kA(k)\cos\frac{(2n+1)\pi k}{L}\right]$	无
对称	偶数	$h[n]=\frac{1}{L}\left[A(0)+2\sum_{k=1}^{\frac{L}{2}-1}(-1)^kA(k)\cos\frac{(2n+1)\pi k}{L}\right]$	A(L/2)=0
反对称	奇数	$h[n]=\frac{2}{L}\sum_{k=1}^{\frac{L-1}{2}}(-1)^{k+1}A(k)\sin\frac{(2n+1)\pi k}{L}$	A(0)=0
反对称	偶数	$h[n]=\frac{1}{L}\left[(-1)^{\frac{L}{2}+n}A(L/2)+2\sum_{k=1}^{\frac{L}{2}-1}(-1)^kA(k)\sin\frac{(2n+1)\pi k}{L}\right]$	A(0)=0

与窗函数法相似，频率抽样法设计滤波器同样存在波动与过渡带宽度之间的平衡问题。如果开始选择的理想滤波器，其通带与阻带间是陡峭的过渡带，通常这样构造的滤波器要比选择平缓过渡带构造的滤波器波动要大。该结论可以由例 5-10 加以说明。 ◀

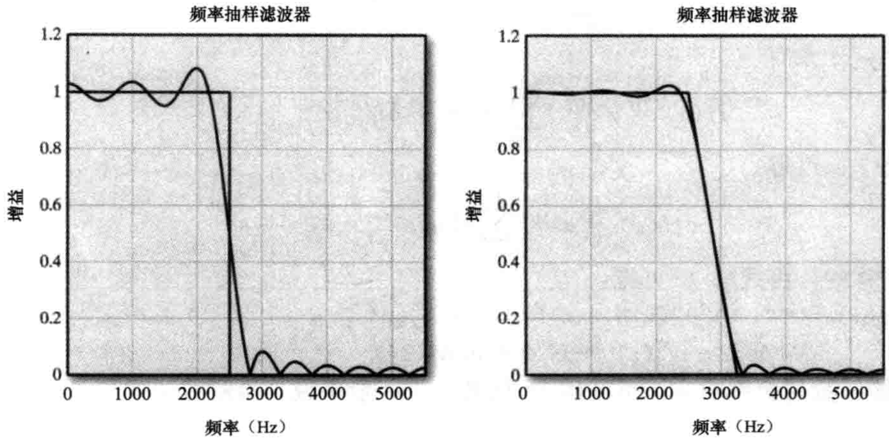
例 5-10 频率抽样法设计滤波器

利用 MATLAB 和频率抽样法设计两个低通滤波器。数字滤波器的长度均为 21，抽样频率为 11025Hz。第一个滤波器的过渡带宽度为 0，通带为 0～2500Hz（阻带为 2500Hz～$f_s/2$Hz）。第二个滤波器的过渡带宽度为 750Hz，因此阻带下限频率由 2500Hz 移到 3250Hz。

解：MATLAB 中采用频率抽样法设计滤波器的函数为 fir2，允许设计时增加窗函数的选择。该例题选择使用矩形窗。频率样本点是在理想滤波器中根据需求数目的抽样而获得。理想滤波器由两个向量描述，构成了一组定义理想滤波器中线段的点集。如下代码构造了两个滤波器，并绘制了它们以及相应理想滤波器的幅值响应。

```
fs = 11025;
L = 21;N = L-1;
fpass = 2500;fstop = 2500;
F = [0 fpass/(fs/2) fstop/(fs/2) 1]; %normalized to fs/2
A = [1 1 0 0];
num = fir2(N, F, A, rectwin(L));
figure(1);
[H f] = freqz(num, 1, 1024, fs);
plot(f, abs(H));                       %first filter
hold on;
plot(F*fs/2, A);                       %Idealized filter
%
fpass = 2500;
fstop = 3250;
F = [0 fpass/(fs/2) fstop/(fs/2) 1];
A = [1 1 0 0];
num = fir2(N, F, A, rectwin(L));
[H f] = freqz(num, 1, 1024, fs);
plot(f, abs(H));                       %second filter
hold on;
plot(F*fs/2, A);                       %Idealized filter
```

结果如图 5.27 所示。



a）由理想低通滤波器构造的20阶FIR滤波器　b）同样是20阶滤波器，但是增加了过渡带从而减小了波动

图 5-27 ◀

与窗函数法相似，频率抽样法允许设计者在波动与过渡带带宽之间有所权衡。总的来说，由表5.7给出的设计公式可认为是通过一组频率样本点拟合其中一条曲线。

5.5 FIR滤波器的帕克斯-麦克莱伦设计技术

虽然到目前为止所讨论的线性相位FIR滤波器的设计方法都相对直接，而且在计算上也没有太多的要求，但都无法得到最优的滤波器。同时，这些方法也不能事先确定滤波器设计时的所有指标，并且期待设计结果满足那些指标。比如，在利用窗函数法设计滤波器时，通常很难获得精确的截止频率。

线性相位FIR滤波器的帕克斯-麦克莱伦设计方法克服了以上这些不足，并且允许设计者设计出某种意义上最优的滤波器[2][3]。这种最优是指所设计的滤波器满足某些设计准则，并且是满足设计准则下的最低阶线性相位FIR滤波器。由于低阶滤波器在应用过程中需要的计算量比较小，因此设计一个低阶滤波器具有重要意义。这也就意味着需要更少的硬件资源，使得更高的抽样率成为可能。

最优意味着定义了某误差函数，此误差函数可用来确定滤波器的性能。误差可以从多种方式进行定义，误差最简单的定义是所设计滤波器的频率响应与理想滤波器频率响应之差。因为误差的符号并不重要，该误差可正可负，所以通常采用绝对值。某些情况下，通带误差对于滤波器的设计比较重要，阻带误差则相对较弱。考虑到这个因素，可以加入权值函数，于是误差可以采用如下定义：

$$E(\Omega) = W(\Omega)\,|\,H_D(e^{j\Omega}) - H(e^{j\Omega})\,|$$

式中，$H_D(e^{j\Omega})$ 表示理想滤波器频率响应，对于低通滤波器，其通带增益是1，阻带增益是0。$H(e^{j\Omega})$ 表示所设计滤波器的频率响应。$W(\Omega)$ 表示权值函数，是关于频率 Ω 的函数。我们希望所设计的滤波器使得这个加权误差的最大值在规定的频带上达到最小，该误差常称为最大最小误差准则或者切比雪夫误差准则。

FIR滤波器的 z 变换可表示成如下形式，

$$H(z) = \sum_{n'=0}^{M} h[n']z^{-n'}\text{，其中，}M\text{ 是滤波器阶数。}$$

如果这是一个线性相位滤波器，$h[n']$ 应是对称的。假定 M 是偶数，将 $H(z)$ 平移M/2个单位，可以得到如下公式：

$$H(z) = \sum_{n'=-M/2}^{M/2} h[n'+M/2]z^{-n'-M/2}$$

提取因子 $z^{-M/2}$，于是有

$$H(z) = z^{-M/2}\sum_{n'=-M/2}^{M/2} h[n'+M/2]z^{-n'}$$

令 $n = n' + M/2$，可得

$$H(z) = z^{-M/2}\sum_{n=0}^{M} h[n]z^{-n+M/2}$$

由于 $h[n]$ 是对称的，由式(5.8)可得，

$$H(z) = z^{-M/2}\{h[0](z^{M/2}+z^{-M/2}) + h[1](z^{M/2-1}+z^{-(M/2-1)}) + \cdots$$
$$+h[M/2-1](z^{-1}+z^{+1}) + h[M/2]\}$$

如需要观察传输函数的频率响应，可利用 $e^{j\Omega}$ 代替 z。一般来说，由欧拉公式，

$$(e^{jx}+e^{-jx}) = 2\cos x$$

频率响应可以写成余弦函数和的形式，

$$h(e^{j\Omega}) = h[M/2]e^{j\Omega M/2} + 2\sum_{n=0}^{M/2} h[M/2]\cos([M/2-n]\Omega) \tag{5.14}$$

不失一般性，对于对称的且对称中心是原点而不是 $M/2$ 的 $h[n]$，式(5.14)变成如下形式，

$$H(e^{j\Omega}) = h[0] + 2\sum_{n=1}^{M/2} h[n]\cos(n\Omega)$$

该式可以写成如下形式：

$$H(e^{j\Omega}) = \sum_{n=0}^{M/2} c_n \cos(n\Omega) \tag{5.15}$$

式中，$c_0 = h[0]$，对于 $n = 1$ 至 $M/2$，有 $c_n = 2h[n]$。

切比雪夫多项式(第 6 章中介绍切比雪夫 IIR 滤波器时将详细地讨论)定义如下：$V_N(x) = \cos(N\arccos x) = n$ 阶 x 的多项式。

如果令 $x = \cos(\Omega)$，切比雪夫多项式变成如下形式：

$$V_N(\cos\Omega) = \cos(N\Omega)$$

式(5.15)改写成：

$$H(e^{j\Omega}) = \sum_{n=0}^{M/2} c_n x^n = P(x)\text{，其中，}x = \cos\Omega\text{。}$$

因此，1 型线性相位偶数阶低通滤波器可以使用三角多项式表示。2 型、3 型与 4 型滤波器也可以得到相似的表示。

交替定理[4]为优化滤波器提供了基础，即对于给定的滤波器阶数，可最小化 $E(\Omega)$。由交替定理，可以规定：

如果 $H(e^{j\Omega})$ 至少具有 $M/2+2$ 个极值点频率 $\Omega_1 < \Omega_2 < \cdots < \Omega_{M/2+2}$，则：

a. 加权误差是以下列形式交替出现的：$E(\Omega_1) = -E(\Omega_2) = E(\Omega_3) = -E(\Omega_4) = \cdots$

b. 最大误差 $|E_{\max}| = |E(\Omega_1)| = |E(\Omega_2)| = \cdots = |E(\Omega_k)|$

此时，在所有频带上，最小化最大加权误差的 $H(e^{j\Omega})$ 是理想滤波器唯一的最佳估计。

根据交替定理，可以得到：

$$E(\Omega_i) = W(\Omega_i)\,|H_D(e^{j\Omega_i}) - H(e^{j\Omega_i})| = \pm |E_{\max}|$$

其中，Ω_i 是极值频率点。

帕克斯-麦克莱伦认为误差可以写成如下形式：

$$E(\Omega_i) = \frac{\sum_{i=1}^{M/2+2} H_D(e^{j\Omega_i})}{\sum_{i=1}^{M/2+2} b_i(-1)^{i+1}/W(\Omega_i)}$$

式中，$b_i = \prod_{\substack{n=1 \\ n\neq i}}^{M/2+2} \frac{1}{\cos(\Omega_i) - \cos(\Omega_n)}$

根据以上信息，帕克斯-麦克莱伦设计了一种寻求最佳滤波器的算法。首先，对于给定的滤波器，估计 $M/2+2$ 个极值频率点的初始位置。经拉格朗日插值法(见例 5-11)利用这些频率点生成经过这些估计频率点的多项式。通过该操作可得到一个连续的且经过所有估计极值频率点的多项式。然而，该多项式的极值点一般不等于估计的极值点。如果不是，则使用插值多项式的极值点来替代估计极值点(如图 5.28 所示)。通过极值点的更新可以得到一个新的多项式，如此迭代，直到上次估计的极值点与生成多项式的极值点吻合，迭代结束。

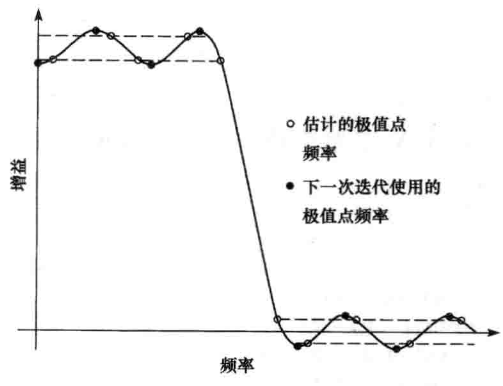


图 5.28　空心圆点表示初始估计的极值点频率。实心圆点表示插值多项式实际的极值点位置。依据雷米兹替代算法，下一次迭代时使用实心圆点作为极值点频率的估计

这种利用插值多项式极值点频率替换估计极值点频率的过程称为雷米兹(Remez)交换算法。

例 5-11 拉格朗日插值

采用拉格朗日插值法产生经过以下各点的多项式：

x	0	0.18	0.4	0.5	0.74	0.83	1.0
y	1.1	0.9	1.1	0.9	0.1	−0.1	0.1

在找到经过这些点的拉格朗日多项式后，移动峰值点的 x 值匹配插值多项式，重新计算一个新的多项式，从而生成一个接近等波动的多项式。

解：拉格朗日插值公式可参见文献[5]

$$P(x)=\sum_{j=0}^{n}y_j\prod_{\substack{k=0\\k\neq j}}^{n}\frac{x-x_k}{x_j-x_k},\text{其中 }P(x)\text{ 经过点}(x_j,y_j),j=0,1,\cdots,n$$

可以直接计算该等式求解 $P(x)$，但是MATLAB通过矩阵形式使得计算更加简单。多项式 $P(x)$ 写成如下形式

$$P(x)=a_0x^n+a_1x^{n-1}+\cdots+a_{n-1}x+a_n$$

其中，a_i 是尝试确定的系数。对于每一个点(x_j ，y_j)可以得到一个等式，所有这些等式的集合写成矩阵形式

$$\begin{aligned}P(x_0)&=a_0x_0^n+a_1x_0^{n-1}+\cdots+a_{n-1}x_0+a_n\\P(x_1)&=a_0x_1^n+a_1x_1^{n-1}+\cdots+a_{n-1}x_1+a_n\\&\cdots\\P(x_n)&=a_0x_n^n+a_1x_n^{n-1}+\cdots+a_{n-1}x_n+a_n\end{aligned}$$

$$\begin{bmatrix}x_0^n & x_0^{n-1} & \cdots & x_0 & 1\\x_1^n & x_1^{n-1} & \cdots & x_1 & 1\\ & & \cdots & & \\x_n^n & x_n^{n-1} & \cdots & x_n & 1\end{bmatrix}\begin{bmatrix}a_0\\a_1\\\cdots\\a_n\end{bmatrix}=\begin{bmatrix}P(x_0)\\P(x_1)\\\cdots\\P(x_n)\end{bmatrix}$$

或者 $\overline{\boldsymbol{X}}\times\overline{\boldsymbol{a}}=\overline{\boldsymbol{P}}$，$\overline{\boldsymbol{a}}=\overline{\boldsymbol{X}}^{-1}\times\overline{\boldsymbol{P}}$ 。

矩阵 $\overline{\boldsymbol{X}}$ 称为范德蒙(Vandermonde)矩阵，可通过MATLAB中的vander函数简单生成。

```
d = .1;
x = [0 .18 .4 .5 .74 .83 1];
y = [1.1 0.95 1.1 0.95 .1 -0.1 0.1];
figure(1);clf;
stem(x, y, 'r');
axis([0 1 -.2 1.3]);
figure(2);clf;
M = vander(x);
a = M\y';
x = 0:.005:1;
P = polyval(a, x);
plot(x, P);
axis([0 1 -.2 1.3]);
xlabel('x');
ylabel('P(x)');
title('Lagrange interpolation');
```

输出结果如图5.29所示。

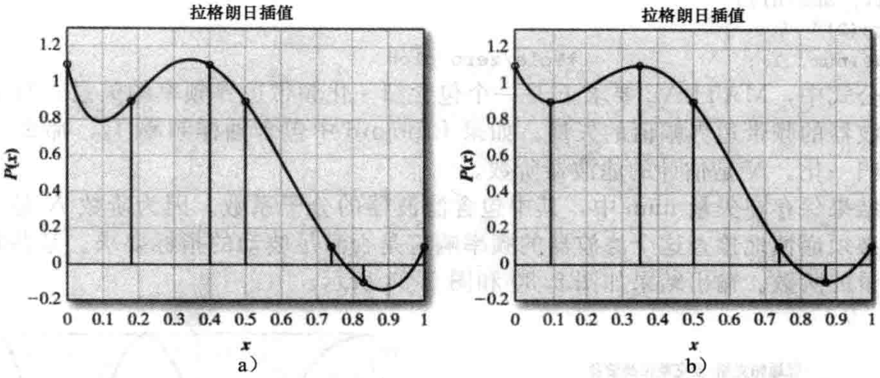


图 5.29 经过 7 个样本点的拉格朗日多项式。MATLAB 程序产生的图 1 与图 2 叠加在一起。图 a 由以上 MATLAB 代码生成。为了匹配图 a 中得到峰值点，重新选取图 b 中的 x 值。新生成的多项式更接近等波动。图 b 中的 x 值是 $x=[0.1.35.74.87.1]$ ◀

总之，帕克斯和麦克莱伦给出了一种线性相位 FIR 滤波器设计方法，该方法基于交替定理与雷米兹交换算法，属于最大最小准则下的优化。帕克斯和麦克莱伦在 1972 年为他们的方法发表了相应的 FORTRAN 算法，并得到了广泛的应用。此算法要求用户输入频带边界参数、波动、权值因子和相应频带内的增益值，采用经验公式估计滤波器阶数的最小值。如下公式可用来估计阶数[6]：

$$N=\frac{-20\lg(\sqrt{\varepsilon_1\varepsilon_2})-13}{14.6\Delta f/f_s}$$

其中，ε_1 和 ε_2 分别是通带与阻带的波动数，Δf 是过渡带宽，f_s 是抽样频率。

此滤波器可以使用阶数的粗略估计来进行设计，通过增大或者减小滤波器阶数寻找满足指标的最小滤波器阶数。帕克斯-麦克莱伦法可利用 MATLAB 完成，函数名是 firpm。函数 firpmord 可用来估计阶数以及加权函数。例 5-12 是这些函数的应用示例。

例 5-12 利用帕克斯-麦克莱伦法设计最优 FIR 低通滤波器

利用帕克斯-麦克莱伦法设计一个最优低通 FIR 线性相位滤波器，滤波器设计指标如下：

抽样频率	11025Hz
通带	0～2300Hz
通带波动	0.03
阻带	3500Hz～$f_s/2$
阻带波动	0.02
权值因子	1

解：输入以下 MATLAB 语句：

```
fs = 11025;
fpass = 2300;fstop = 3000;
rpass = 0.03;rstop = 0.02;
F = ([fpass fstop]);            %define the frequency bands
M = ([1 0]);                    %gain within the bands
Err = [rpass rstop];            %Ripple in the bands
%W is the weighting function within each band
[N F A W] = firpmord(F, M, Err, fs);
num = firpm(N, F, A, W);
figure(1)
[H f] = freqz(num, 1, 1024, fs);
```

```
plot(f, abs(H));
figure(2)
zplane(num, 1);                    %Pole/zero plot
```

以上公式中，MATLAB要求 F 是一个包含归一化频带边界频率的矢量，M 是一个包含理想滤波器的频带边界幅值的矢量。如果 firpmord 中包含抽样频率 f_s，那么 F 中的频率不需要归一化。N 是估计的滤波器阶数。

返回结果保存在矢量 num 中，其中包含滤波器的分子系数。因为阶数 N 是一个估计值，所以必须谨慎地检查这个滤波器的频率响应是否满足波动的指标要求。某些情形下需要增加少量的阶数。输出结果如图 5.30 和图 5.31 所示。

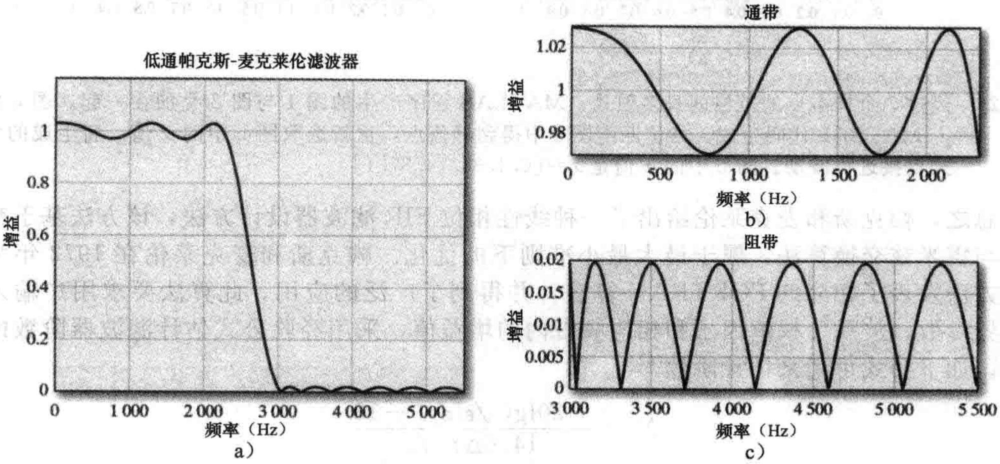


图 5.30　满足例中设计指标的帕克斯-麦克莱伦滤波器的幅度谱

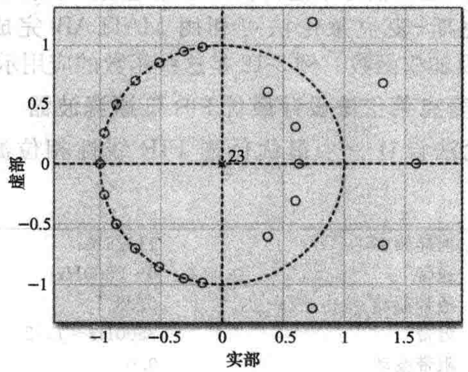


图 5.31　23 阶滤波器的 z 平面零极点图

例 5-13　利用帕克斯-麦克莱伦法设计最优带阻 FIR 滤波器

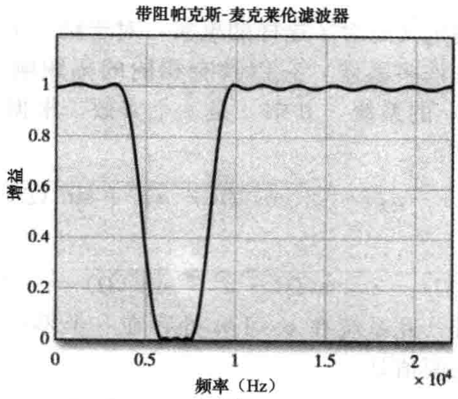
利用帕克斯-麦克莱伦法设计一个最优带阻 FIR 线性相位滤波器，滤波器设计指标如下：

抽样频率	44100Hz
通带 1	0～3600Hz
通带 1 波动	0.03
阻带	5820～9600Hz
阻带波动	0.02
通带 2	7600Hz～$f_s/2$
通带 2 波动	0.03

解： 此题的 MATLAB 程序除了前几行外，其余与例 5-12 相似。新的代码与输出结果如图 5.32 与图 5.33 所示。

```
fs = 44100;
fps1 = 3600;fps2 = 9600;
fst1 = 5820;fst2 = 7600;
rpass = 0.01;rstop = 0.01;
F = ([fps1 fst1 fst2 fps2]);
M = ([1 0 1]);
Dev = [rpass rstop rpass];
[N F A W] = firpmord(F, M, Dev, fs);
num = firpm(N, F, A, W);
```

a)



b)

图 5.32 利用帕克斯-麦克莱伦算法设计的等波动带阻滤波器

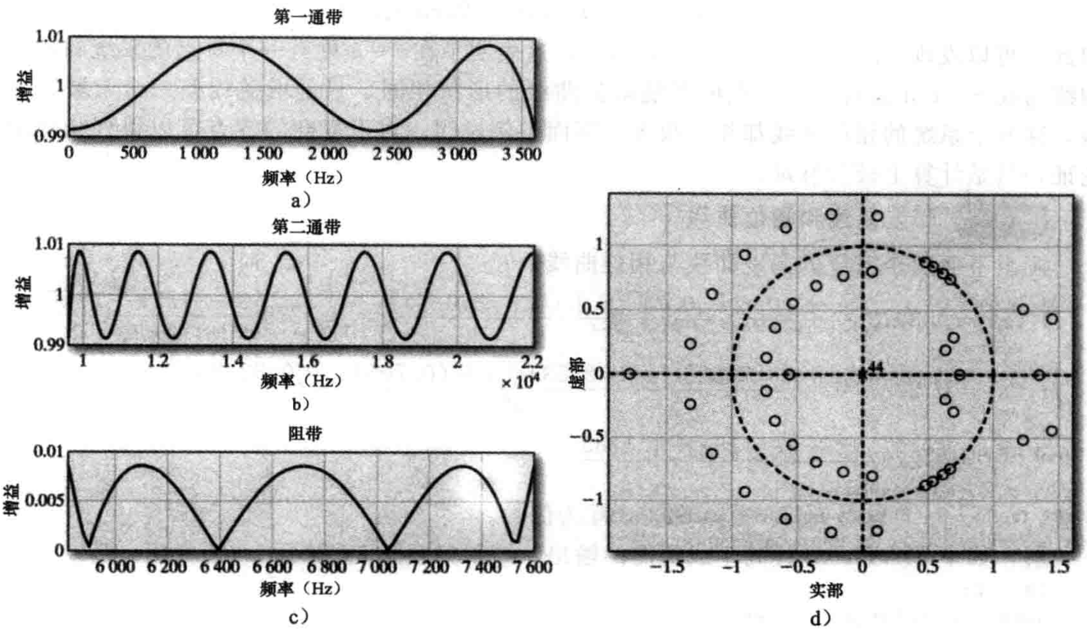


图 5.33 图 a、b、c 分别是例 5-13 所设计的满足指标的滤波器对应的通带与阻带波动。图 d 是 44 阶 FIR 滤波器的零极点图

5.6　最小相位FIR滤波器

通过可视化系统传输函数零极点位置对其相位曲线的影响，可以很简易地给出系统相位特征的几何解释。

最小相位系统——对于给定的幅频响应，如果某系统的相位曲线相对于零相位的偏差是最小的，那么这个系统是最小相位系统。

最大相位系统——对于给定的幅频响应，如果某系统的相位曲线相对于零相位的偏差是最大的，那么这个系统是最大相位系统。

混合相位系统——如果频率从0Hz至$f_s/2$范围，如果某系统的相位曲线相对于零相位的偏差介于最小相位系统与最大相位系统对应的误差之间，那么这个系统是混合相位系统。

最小与最大相位系统的定义隐含了这样的事实，对于给定的幅频曲线，存在多条相位曲线。换句话说，可以存在许多系统，它们具有相同的幅频响应，但具有不同的相位响应。考虑一个具有零点$z=a$的系统，其中a是一个实数。根据传输函数在单位圆上的求值，计算这个零点的幅值是，

$$|z-a| = \sqrt{(\cos(\Omega)-a)^2+\sin^2(\Omega)}$$

展开等式得到：

$$|z-a| = \sqrt{\cos^2(\Omega)-2a\cos(\Omega)+a^2+\sin^2(\Omega)} = \sqrt{1-2a\cos(\Omega)+a^2}$$

现在再考虑第二个系统，此系统在$z=1/a$处存在一个零点。根据传输函数在单位圆上的求值，计算这个零点的幅值是，

$$|z-1/a| = \sqrt{1-(2/a)\cos(\Omega)+1/a^2}$$

等式写成如下形式，

$$|z-1/a| = (1/a)\sqrt{a^2-2a\cos(\Omega)+1}$$

因此，可以发现$|z-1/a|=(1/a)|z-a|$，这表明了在$z=a$处有一个零点的系统的幅值曲线与在$z=1/a$处有一个零点的系统幅值曲线的形状相同，只是增益相差一个常数。然而，这两个系统的相位曲线却相差很大，下面举例说明。对于复杂的零点可以得到相似的论证，只是计算上较为麻烦。

例5-14　幅度曲线和相位曲线

画出下面两个系统的幅度曲线与相位曲线

$$H_1(z)=0.3279\frac{[z+(0.6+j0.7)][z+(0.6-j0.7)]}{z^2}=0.3279\frac{z^2+1.2z+0.85}{z^2}$$

$$H_2(z)=0.2787\frac{[z+(0.70588+j0.82353)][z+(0.70588-j0.82353)]}{z^2}$$

$$=0.2787\frac{z^2+1.4118z+1.1765}{z^2}$$

注意，$0.6\pm j0.7$与$0.70588\pm j0.82353$互为倒数。

解：如下MATLAB代码生成所需的输出，结果如图5.34所示。

```
fs = 2;
num1 = 0.3279*[1 1.2 .85];
num2 = 0.2787*[1 1.4118 1.1765];
den = [1 0 0];
[H1 F] = freqz(num1,den,512,fs);
```

```
subplot(2,3,1)
 plot(F,abs(H1))
subplot(2,3,2)
 Theta = unwrap(angle(H1))*(180/pi);
 plot(F,Theta)
subplot(2, 3, 3);
 zplane(num1, den);
%
[H2 F] = freqz(num2,den,512,fs);
subplot(2,3,4)
 plot(F,abs(H2))
subplot(2,3,5)
 Theta = unwrap(angle(H2))*(180/pi);
 plot(F,Theta)
subplot(2, 3, 6);
 zplane(num2, den);
```

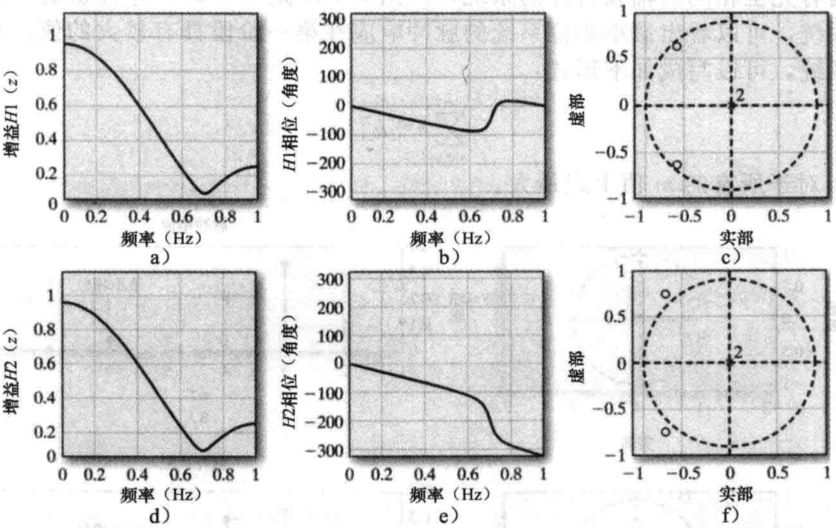


图 5.34 零点(复数零点)互为倒数的两个系统的幅频图和相频图。两个系统的幅频图仅仅是增益不同，而相频图差别却很大。在图 a 中，$H_1(z)$是最小相位系统，在图 d 中 $H_2(z)$是最大相位系统◀

例 5-14 说明，如果两个传输函数的零点是互为倒数，则它们的幅频曲线相同(这个结论同样可以适用于极点，只不过倒数极点可能导致系统不稳定)。因此，如果系统具有 N 个不在单位圆上的复数零点对和M个不在单位圆上的实数零点，就可以得到 2^{M+N}个具有同样幅频响应的传输函数，但它们的相位谱不相同。由于相位平移与时间延迟相一致，因此最小相位系统具有最小的时延。所以在一些系统中，知道哪个传输函数对应的系统可能具有最小的相位是重要的。

考虑一个二阶系统，其传输函数中含有 1 个复数零点对和两个位于原点的极点，如图 5.35所示。图 5.35a 显示的是两个零点(都在 z 平面单位圆内)和两个极点(在原点)。使用频率响应的几何插值，可以发现对于零点均出现在单位圆内的情况，当频率从 0Hz 变化到 $f_s/2$ 时，角度总共变化了 $\theta_1+\theta_2=$

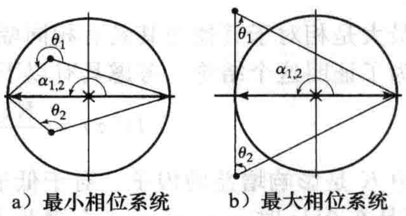


图 5.35 此图显示了由两个零点引起的总的角度变化。a)两个零点均位于单位圆内，b)两个零点均位于单位圆外。对于图 a，$\theta_1+\theta_2=2\pi$，但是对于图 b，$\theta_1+\theta_2=0$

360°。对于两个极点，总的角度变化为 $\alpha_1+\alpha_2=360°$。因为极点角度减去零点角度，所以在系统(a)的相位响应中角度改变是 0°。因此，该系统的相位响应应该是从 0° 开始而终止于 0°。对于系统(b)，其两个零点(其为系统(a)中零点的倒数)，都在单位圆外，可以发现对于零点总的角度变化为 $\theta_1+\theta_2=0°$，而对于极点角度变化仍然是 360°。这样系统(b)的相位响应中角度改变是 $-360°$。

总的来说，所有零点都位于单位圆内的系统为*最小相位系统*，所有零点位于单位圆外的系统为*最大相位系统*。如果既有单位圆内又有单位圆外的零点，那么这样的系统称为*混合相位系统*。零点出现在单位圆上导致相位曲线的不连续。由于在单位圆上的零点有相应倒数，且是它们的共轭对，因此这些零点对于判断系统最小相位还是最大相位并不起作用。

当特定频率的信号通过系统时，系统相位的改变转变成信号的时间延迟，因此设计一个最小相位系统往往很重要。最小相位系统的另一特征体现在脉冲响应上。图 5.36 显示的是两个具有完全相同的幅频特性的系统，但图 5.36a 是一个最小相位系统，图 b 是一个最大相位系统，可以看出最小相位系统的脉冲响应在第一阶段具有较大的值。对于任意的最小相位系统，可以写成如下形式：

$$\sum_{n=0}^{m} h^2[n]$$

从 0 至 N，对于所有的 m 值上式最大。

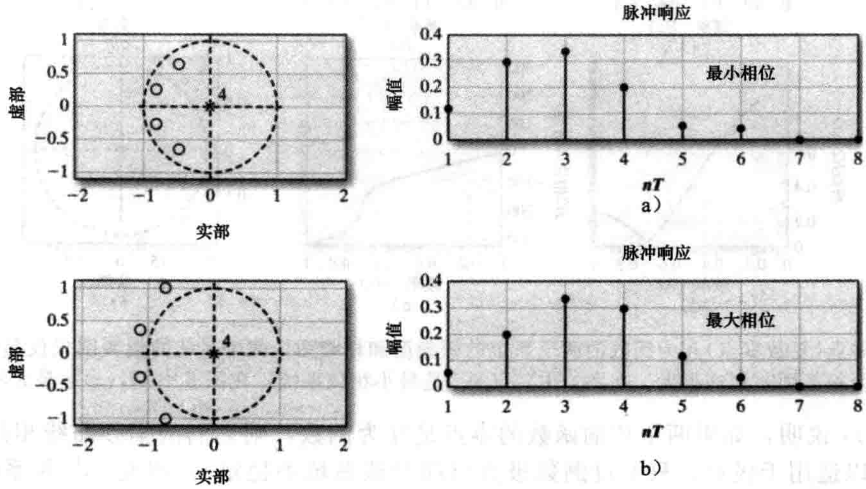


图 5.36　a)最小相位系统及其对应的脉冲响应。b)最大相位系统及其对应的脉冲响应。a)和 b)具有相同的幅值响应

最大是相对于其他与其具有相同幅度响应的系统而言。

为了证明这个结论，考虑具有以下传输函数的 FIR 滤波器：

$$H(z)=\frac{K(b_0z^N+b_1z^{N-1}+\cdots+b_N)}{z^N}$$

等式中 K 是影响增益的因子。对于低通滤波器，增益因子的选择应保持在 0Hz 时增益为 1，但是在 0Hz 时 $z=e^{j0}=1$，因此 K 可以由以下等式计算：

$$K=\frac{1}{b_0+b_1+\cdots+b_N}$$

如果从 $H(z)$ 提取一个实数零点项，剩下的函数称为 $R(z)$，可以得到如下关系：

$$H(z)=K_RR(z)K_z(z-z_k)/z$$

式中，K_R 是 $R(z)$ 的增益因子，K_z 是提取零点项的增益因子，并且 $K = K_R K_z$。对于一个实零点信号，K_z 的值是 $|1/(1-z_k)|$。因此 $H(z)$ 可以进一步写成：

$$H(z) = K_R R(z)\left|\frac{1}{1-z_k}\right|(1-z^{-1}z_k) = H_1(z)\cdot H_2(z)$$

式中，$H_1(z) = K_R R(z)$，$H_2(z) = |1/(1-z_k)|(1-z^{-1}z_k)$。

当 z_k 由其倒数所替换时，根据该等式，考虑所构造的非最小相位系统将有什么变化。因此，构造第二个传输函数 $G(z)$，它将 $H(z)$ 中 z_k 处的零点代替为其倒数。可以写成如下形式：

$$G(z) = KR(z)(z-1/z_k)/z = K_R R(z) K_{1/z}(z-1/z_k)/z$$

其中，$K_{1/z}$ 是倒数零点的增益因子。$K_{1/z} = |1/(1-1/z_k)| = |z_k/(z_k-1)|$，因此 $G(z)$ 可以改写成如下形式：

$$G(z) = K_R R(z)\left|\frac{z_k}{z_k-1}\right|(1-z^{-1}/z_k) = G_1(z)G_2(z)$$

其中，$G_1(z) = H_1(z) = K_R R(z)$，$G_2(z) = |z_k/(z_k-1)|(1-z^{-1}/z_k)$。

对比 $G(z)$ 和 $H(z)$ 可以发现，这两个传输函数中都含有 $K_R R(z)$ 项。因此可以比较提取出来的两个零点项的影响。这两个等式的脉冲响应为：

$$\left|\frac{1}{1-z_k}\right|(1-z^{-1}z_k) \Leftrightarrow h_2[n] = \left|\frac{1}{1-z_k}\right|\{1-z_k \quad 0 \quad 0\cdots\}$$

$$\left|\frac{z_k}{z_k-1}\right|(1-z^{-1}/z_k) \Leftrightarrow g_2[n] = \left|\frac{z_k}{z_k-1}\right|\{1-1/z_k \quad 0 \quad 0\cdots\}$$

由上式可以看出如果 $|z_k| < 1$，则

$$\sum_{n=0}^{m}|h_2[n]| \geqslant \sum_{n=0}^{m}|g_2[n]|$$

对于复数零点，虽然计算上比较繁琐，但可以得到相似的结果。

由于 $H(z) = H_1(z)\cdot H_2(z)$，因此必须通过卷积计算 $h_1[n] * h_2[n]$ 得到 $h[n]$。因此可以得到如下结论：

$$E = \sum_{n=0}^{m} h^2[n]$$

当 $H(z)$ 为最小相位系统时，E 达到最大。有时称 E 为系统的*局部能量*。局部能量等式的实际结果来源于最小相位系统的阶跃响应。由于系统的阶跃响应 $S(n)$ 可以写成脉冲响应的形式 $S(n) = h(n) + S(n-1)$，可以发现，具有最大化局部能量的系统具有最快的时间常量。

5.7 应用

5.7.1 滑动平均 FIR 滤波器

有限序列的平均值如下所示：

$$\text{avg} = \frac{1}{N}\sum_{i=1}^{N} x_i$$

当 N 很大时该公式很难应用，因为在输出结果之前它要求所有 x_i 的值均为已知。在不知道 N 值因而难以确定求和上限的时候，该公式很难实现。实际中计算平均值往往使用滑动平均，滑动平均是最近的 N 个输入项的平均，由下式给出：

$$A_M[k] = \frac{1}{N}\sum_{i=k-N+1}^{k} x_i \tag{5.16}$$

例 5-15 滑动平均

计算出下面序列的滑动平均序列：

$$x=[0,1,1,1,2,2,2,4,1,5,7,4,6,8,3,7,1,\cdots]$$

设定 $N=3$ 。

解：由式(5.16)计算结果如下：

$$x=[0,1,1,1,2,2,2,4,1,5,7,4,6,2,3,7,1,\cdots]$$

$$A_{\mathrm{M}}[k]=[0,1/3,2/3,1,4/3,5/3,2,8/3,7/3,10/3,13/3,16/3,17/3,4,11/3,4,11/3,\cdots]$$ ◀

可以很简单地求解式(5.16)对应的传输函数。当 $N=3$ 时，式(5.16)变成如下形式：

$$A_{\mathrm{M}}[k]=\frac{1}{3}(x[k]+x[k-1]+x[k-2])$$

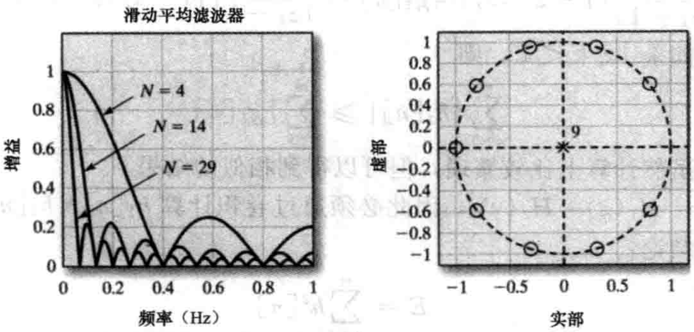
传输函数为：

$$H_{\mathrm{M}}(z)=\frac{z^2+z+1}{3z^2}$$

总的来说，对于一个长度为 L 的滑动平均系统

$$H_{\mathrm{M}}(z)=\frac{z^{L-1}+z^{L-2}+\cdots+1}{Lz^{L-1}} \tag{5.17}$$

式(5.17)称为滑动平均系统的*精确非递归形式*。滑动平均系统是传输函数的零点全部位于单位圆上的 FIR 低通滤波器。图 5.37 给出了不同阶数的滑动平均系统的幅频图以及零极点分布图。



a）阶数分别是4、14和29的滑动平均滤波器的幅频响应　　b）9阶滑动平均滤波器的零极点分布图

图 5.37

当数据多次通过该滤波器时会发生什么，是滑动平均系统的一个很有趣的特征。我们称这样的系统为*多重滑动平均滤波器*(MAF)。例如，噪声数据通过一个长度为 3 的滑动平均系统，接着再将此滑动平均滤波器的输出继续通过第二个相同的滑动平均滤波器，这样便得到了一组滑动平均数据的滑动平均值。在频域中，此系统的传输函数等于传统滑动平均滤波器的平方。对于长度为 3 的滤波器，传输函数如下：

$$\mathrm{mp2Avg}=\left(\frac{z^2+z+1}{3z^2}\right)^2=\frac{z^4+2z^3+3z^2+2z+1}{9z^4}$$

分析该传输函数可以发现，其与三角窗函数的传输函数相同。如果继续处理，可以发现 4 重滑动平均滤波器与等长度的布莱克曼窗函数的频率响应相似。

例 5-16 滑动平均滤波器

构造一个长度为 8 的滑动平均滤波器，并利用所设计的滤波器生成一个 4 重滑动平均滤波器，并比较其与等长度的布莱克曼窗函数的频率响应。

解：如下 MATLAB 代码可以产生一个 32 阶布莱克曼窗函数与一个 32 阶 4 重滑动平均滤波器的幅值响应图，结果如图 5.38 所示。

```
fs = 2;                          %Sample frequency
N = 8;
ma1 = ones(1, N)/N;              %The original moving avg filter
ma2 = conv(ma1, ma1);            %Convolve the moving avg filter with
                                 % itself to get a 2-pass MAF
ma4 = conv(ma2, ma2);            %Convolve the 2-pass with itself for
                                 % a 4-pass MAF
[Hma4 f] = freqz(ma4, den, 1024, fs);
figure(1);clf;
Hma4 = abs(Hma4)/max(Hma4);      %normalize and change to db
Hma4DB = 20*log10(Hma4);
plot(f, Hma4DB);
hold on;
wBlk = blackman(length(ma4));%Create a Blackman window
[Hb f] = freqz(wBlk, den, 1024, fs);
Hb = abs(Hb)/max(Hb);
HbDB = 20*log10(Hb);             %Normalize and change to db
plot(f, HbDB, 'r');
```

相比之下，布莱克曼窗有比较好阻带性能，而且即使随着阶数的增加，滑动平均系统相比较等长度布莱克曼窗，其峰值波动较大。◀

多重滑动平均滤波器可用于估计布莱克曼窗的频率响应。在实时系统中，滑动平均估计相比于直接使用布莱克曼窗处理数据需要的计算量较小。

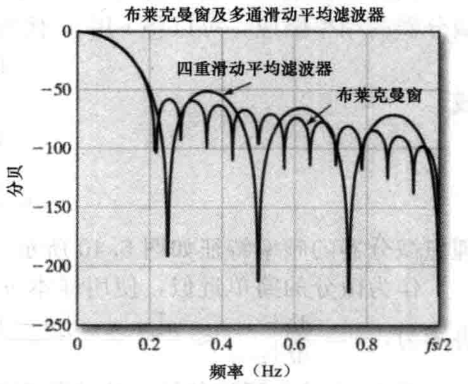


图 5.38　比较长度为 32 的布莱克曼窗与等长度的 4 通滑动平均滤波器的幅度响应

5.7.2　梳状滤波器

使用滑动平均滤波器构造梳状滤波器有两种不同的方法。第一种方法，在 FIR 滑动平均滤波器的传输函数中，于 $z=+1$ 处增加一个零点可构造梳状滤波器。对于 N 阶 FIR 滤波器，这将构造出具有 $N/2$ 个等间隔主瓣的梳状滤波器。第二种方法，利用滑动平均滤波器的幅频图，在频域内进行尺度变换使得主瓣在频率空间内复制 M 次，从而得到具有 M 个“齿”的梳状滤波器。第二种方法得到的是 IIR 滤波器，将在 6.7 节中进行详细讨论。

FIR 滑动平均滤波器的传输函数已在(5.17)中给出，

$$H_M(z)=\frac{z^{L-1}+z^{L-2}+\cdots+1}{Lz^{L-1}}$$

传输函数在 $z=+1$ 处增加一个零点，得到：

$$H_M(z)=\frac{z^{L-1}+z^{L-2}+\cdots+1}{Lz^{L-1}}\,\frac{z+1}{z}=\frac{z^L-1}{Lz^L}$$

因此，梳状滤波器的传输函数为：

$$H_C(z)=K\frac{z^N-1}{z^N}\tag{5.18}$$

图 5.39 是典型的 10 阶梳状滤波器。如果滤波器的阶数是奇数，那么在 $z=-1$ 处没有零点，而且在 $f_s/2$ 处的增益是 1。$N/2$ 瓣的中心是 $kf_s/N, k=0,1,\cdots,N/2$，N 为偶数。

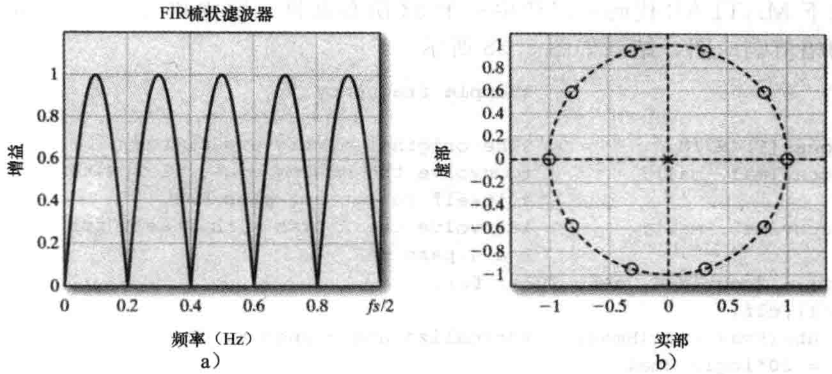


图 5.39　通过对滑动平均滤波器增加一个零点而构造的梳状滤波器，其阶数为 10

通过将极点沿着原点径向移动到单位圆附近，并以零点为间隔，可实现梳妆滤波器瓣的尖锐化。当然，这也使得滤波器变化成 IIR 滤波器，而且增加了计算量。

5.7.3　微分器

微分器实现输入信号的时域微分。信号时域微分相当于拉普拉斯域乘以 s，为了得到微分器的频率响应，可以将 s 用 $j\omega$ 代替，

$$H_{diff}(\omega)=j\omega$$

或

$$|H_{diff}(\omega)|=\omega$$
$$\angle|H_{diff}(\omega)|=\frac{\pi}{2} \tag{5.19}$$

理想微分器的频率特征如图 5.40 所示。

作为微分的简单近似，使用样本 n 与样本 $n-1$ 之间的斜率作为微分器 n 时刻的值，表示为，$y=\left.\frac{\mathrm{d}u}{\mathrm{d}t}\right|_{t=nT}\approx\frac{u[n]-u[n-1]}{T}$，或者利用 z 变换$\frac{y(z)}{u(z)}=\frac{z-1}{Tz}$。

图 5.41 给出了该传输函数的频率响应，并与理想微分器进行了对比。从该图可以清晰看出，近似微分器在低频时比较理想，但当频率达到 $f_s/2$ 时，误差将十分明显，此在意料之中，原因在于高频部分信号样本间的变化较大。该方法所要求的计算量小，适合低频信号微控制器，例如，温度或湿度传感器等。

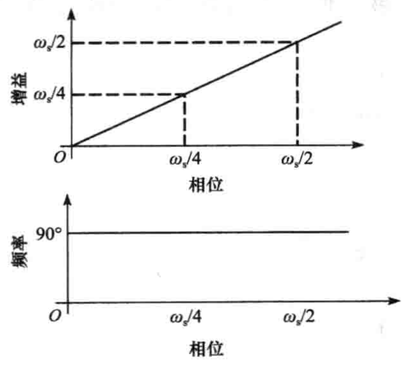


图 5.40　理想微分器的幅频与相频响应

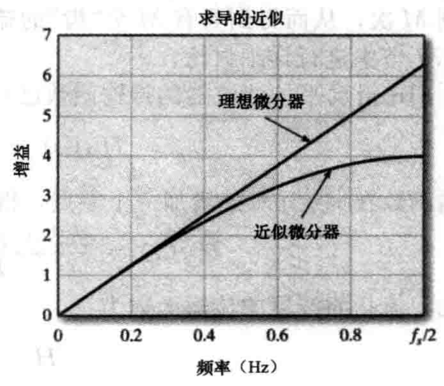


图 5.41　近似微分器频率响应在 nT 点与$(n-1)T$点之间的斜率

为了得到较好的近似微分器，可利用傅里叶级数方法，如有必要也可加入窗函数，

MATLAB 中的 firls 函数可实现此方法。如果加入"differentiator"参数，那么生成的微分器将在低频误差处分配较重的权值。相似地，帕克斯-麦克莱伦法和频率抽样法也可用于构造近似微分器[7]。

例 5-17 理想微分器的数字滤波器近似

利用傅里叶级数方法，设计一个数字滤波器用于近似理想微分器，抽样频率为 2Hz。

解： 下面的 MATLAB 代码使用 firls 函数构造微分器。

```
fs = 2;
N = 11;     %Order
F = [0 1]; %Frequency points
M = [0 1]; %Corresponding magnitude points
num = 2*pi*firls(N, F, M, 'differentiator');
den = 1;
[Hdf f] = freqz(num, den, 1024, fs);
plot(f, abs(Hdf));
```

图 5.42 显示了使用 firls 函数设计的不同阶数的滤波器。注意，当滤波器阶数是偶数时，这种方法会在 $f_s/2$ 产生一个零点。◀

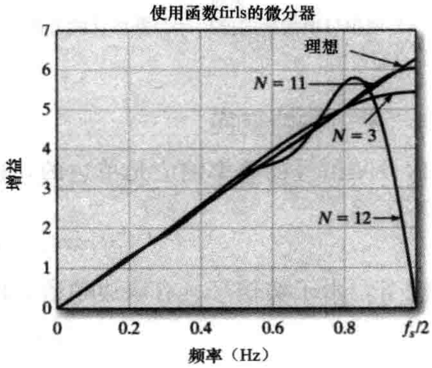


图 5.42 利用 MATLAB 中的 firls 函数设计三个微分器，其中直线是理想微分器的频率响应

微分器在许多不同的系统中得到了广泛使用，尤其是在数字控制系统中，常用作网络补偿和 PID 控制器部件。

例 5-18 例 5-17 所设计滤波器的应用

在第 3 章介绍傅里叶级数时，方波可以由下式近似：

$$x(t) = \sum_{\substack{k=1 \\ k\text{为奇数}}}^{N} \frac{4}{k\pi}\sin(k\omega_0 t)$$

利用 MATLAB 构造 $x(t)$，其中，$\omega_0 = 200\pi$，f_s=22050Hz，N=21。使用 1024 个样点。利用 MATLAB 中的 firls 函数构造例 5-17 中的 11 阶微分器，对该信号进行滤波并画出结果。

解： 首先由例 5-17 使用 f_s=22050Hz 构造微分器。

```
fs = 22050;
N = 11;     %Order
F = [0 1]; %Frequency points
M = [0 1]; %Corresponding magnitude points
num = 2*pi*firls(N, F, M, 'differentiator');
den = 1;
```

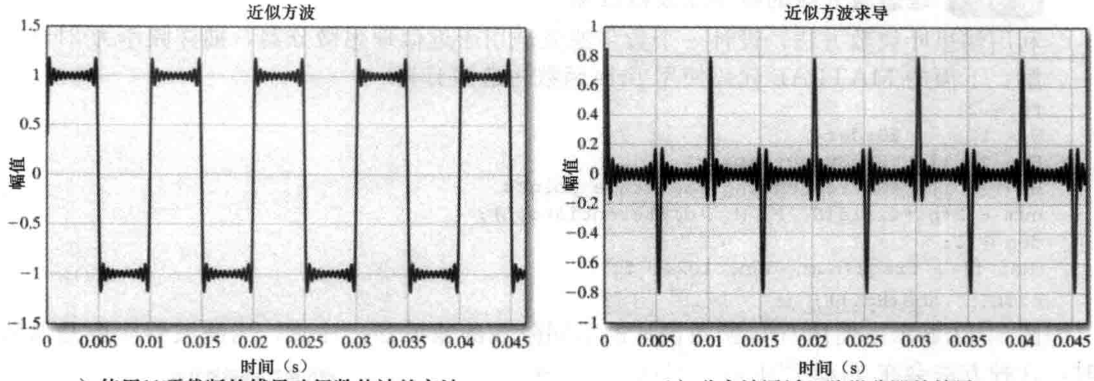
下面生成近似的方波。

```
T = 1/fs;
n = 1:1024;
nT = n*T;
w = 200*pi;
sq = zeros(1, length(nT));
for k = 1:2:21
    sq = sq + (4/(k*pi))*sin(k*w*nT);
end
figure(1);clf;
plot(nT, sq);
```

最后使用滤波器对近似的方波进行微分。

```
sqDiff = filter(num, den, sq);
figure(2);clf;
plot(nT, sqDiff);
```

结果如图 5.43 所示。



a）使用11项截断的傅里叶级数估计的方波　　b）此方波通过11阶微分器的结果

图 5.43

5.7.4 希尔伯特变换

假设某信号的频率响应是单边的，也就是说，信号 $x[n]$ 的 DTFT 为 $X(e^{j\Omega})$，于是得到如下关系

$$X(e^{j\Omega}) = 0, \quad \omega_s/2 \leqslant \omega \leqslant \omega_s$$

该情况下，由于实信号具有双边频谱，因此该信号的时域将是复数形式，即

$$x[n] = x_R[n] + jx_I[n]$$

而且

$$X(e^{j\Omega}) = X_R(e^{j\Omega}) + jX_I(e^{j\Omega}) = 0, \quad \omega_s/2 \leqslant \omega \leqslant \omega_s$$

等式可写为

$$X_R(e^{j\Omega}) = -jX_I(e^{j\Omega}), \quad \omega_s/2 \leqslant \omega \leqslant \omega_s$$

可求解出 X_I 为，

$$X_I(e^{j\Omega}) = jX_R(e^{j\Omega}), \quad \omega_s/2 \leqslant \omega \leqslant \omega_s$$

$x_R[n]$ 与 $x_I[n]$ 都是实数序列，这就意味着 $x_R[n]$ 与 $x_I[n]$ 的离散时间傅里叶变换以共轭对形式出现，以便使

$$X_I(e^{j\Omega}) = -jX_R(e^{j\Omega}), \quad 0 \leqslant \omega \leqslant \omega_s/2$$

单边信号这种实部与虚部之间的关系称为希尔伯特变换，将 $x_R[n]$ 转变 $x_I[n]$ 的滤波器称为希尔伯特变换器或者希尔伯特滤波器。

希尔伯特变换器的幅频响应为 1，是使相位滞后 90° 的全通移相装置。希尔伯特变换在通信领域信号单边带幅度调制中得到了广泛的应用。

理想希尔伯特变换器的频率响应可以写成如下形式，

$$H_H(e^{j\Omega}) = \begin{cases} -j, & 0 \leqslant \omega \leqslant \omega_s/2 \\ +j, & \omega_s/2 \leqslant \omega \leqslant \omega_s \end{cases}$$

其对应的 IDTFT 为

$$h[n] = \begin{cases} \dfrac{2}{\pi n T}, & n \text{ 是偶数且 } n \neq 0 \\ 0, & \text{其他情况} \end{cases} \tag{5.20}$$

正如理想低通滤波器，希尔伯特变换器不可能构造成数字滤波器。然而，可以利用 FIR 滤波器近似希尔伯特变换器。

例 5-19 利用 FIR 滤波器近似希尔伯特变换器

利用式(5.20)设计 FIR 滤波器以近似希尔伯特变换，$N=30$。

解：由 MATLAB 根据式(5.20)生成包含有 31 项的 FIR 滤波器系数，以近似希尔伯特变换器。图 5.44 给出了该近似变换器的幅频图和相频图。

```
fs = 2; T = 1/fs; % Sample frequency = 2
N = 30;           % Order = 30
a = zeros(1, N+1);
for k = -N/2:2:N/2
    if(k ~= 0)
       a(k+N/2+1) = 2/(pi*k*T);
    end
end
[H f] = freqz(a, 1, 1024, fs);
Hnorm = abs(H)/max(abs(H));
plot(f, Hnorm);
```

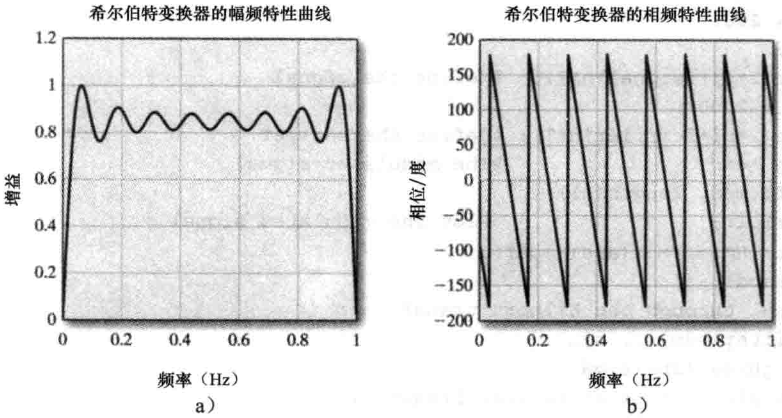


图 5.44　希尔伯特变换器的幅频图和相频图 ◀

同样可以结合帕克斯-麦克莱伦设计法，并使用 MATLAB 中的 firpm 函数构造希尔伯特变换器。firpm 函数需要设定 3 个变量 N、F、M。N 为滤波器阶数。F 和 M 包含滤波器频率响应中指定点幅度的向量。F 要求是归一化的，因此在 $f_S/2$ 处等于 1。M 是 F 中相应频率点的幅值。下面三行 MATLAB 代码用于构造 30 阶帕克斯-麦克莱伦变换器。

```
F = [.1 .9];
M = [1 1];
h = firpm (30, F, M, 'hilbert');
```

通信系统中单边带信号的产生便是希尔伯特变换器应用的实例。考虑频率为 Ω_S 的低频信号，调制频率为 Ω_c 的高频载波，得到的信号表示为，

$$f(t)=(1/2)\cos(\omega_S t)\cos(\omega_c t)=\cos[(\omega_c+\omega_S)t]+\cos[(\omega_c-\omega_S)t]$$

调制后信号 $f(t)$ 既包含上边带也包含下边带。由于通过解调上边带或下边带都可以恢复出原信号，因此无需传输两个边带。一种方法是利用低通或者高通滤波器滤除其中的一个边带。更有效的方法是使用希尔伯特变换器。

考虑信号 $s[n]=e^{j\Omega_S n}=s_r[n]+js_i[n]$ 和载波信号 $c[n]=e^{j\Omega_c n}=\cos(\Omega_c n)+j\sin(\Omega_c n)$。两个信号相乘得到调制后的信号为

$$f[n]=f_r[n]+\mathrm{j}f_i[n]=s_r[n]\cos(\Omega_c n)-s_i[n]\sin(\Omega_c n)+\mathrm{j}(s_i[n]\cos(\Omega_c n)-s_r[n])\sin(\Omega_c n)$$

利用希尔伯特变换，可以由信号的实部生成虚部，图5.45显示了如何产生调制后信号实部的过程。

例5-20　利用希尔伯特变换器产生信号的单边带信号

按照图5.45框图，利用MATLAB仿真单边带信号的产生。假定信号是200Hz的正弦信号，载波信号为频率2000Hz的正弦信号。抽样频率为11025Hz，利用firpm函数，以及20阶希尔伯特变换器进行仿真。

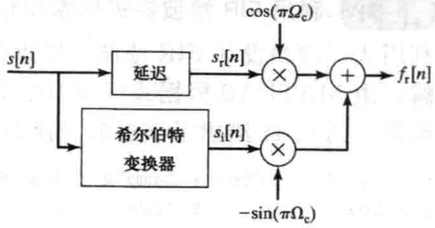
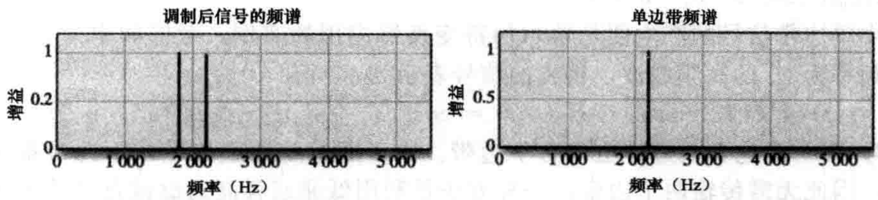


图5.45　希尔伯特变换器生成信号的单边带信号

解：下面的MATLAB代码产生2kHz载波，并用200Hz信号对其调制。根据图5.45框图生成单边带信号，结果见图5.46。

```
fs = 11025; T = 1/fs;  % Sample frequency = 1000
N = 20;                % Order = 20
F = [.1 .9];M = [1 1];
num = firpm(N, F, M, 'hilbert'); %Create the Hilbert transformer
[H f] = freqz(num, 1, 1024, fs);
fsignal = 200;
n = 0:fs;
sig = sin(2*pi*fsignal*n*T);  %define the signal
fcarrier = 2000;
car = sin(2*pi*fcarrier*n*T); %define the carrier
mod = sig.*car;               %The modulated signal
Hmod = fft(mod, length(n));
figure(1);clf;                %Plot the modulated signal
Hmod = abs(Hmod)/max(abs(Hmod));
plot(n, Hmod);
%Run signal through the Hilbert transformer
sigH = filter(num, 1, sig);
sigH = sigH/max(abs(sigH));
%Define a sin and cos at carrier frequency
carCos = cos(2*pi*fcarrier*n*T);
carSin = sin(2*pi*fcarrier*n*T);
% Delay the signal by N/2
sig = [zeros(1, N/2) sig(1: length(sig)-N/2)];
%Form the single sideband signal
modH = sig.*carSin + sigH.*carCos ;
HH = fft(modH, length(n));
HH = abs(HH)/max(abs(HH));
figure(2);clf; %Plot the spectrum of the single
plot(n, HH);   % sideband signal
```



a）2kHz载波信号被200Hz正弦信号调制后的信号频谱　　b）由图5.45所示的希尔伯特变换器生成调制后信号的单边带信号

图　5.46

5.8 FIR 滤波器小结

IIR 滤波器存在反馈回路而 FIR 滤波器没有，这个差别所具有的优点和缺点需要在设计中权衡。其重要的特征差别归纳如下。

脉冲响应：不管其实现精度，在有限的时间之后，FIR 滤波器的脉冲响应等于零。

稳定性：FIR 滤波器没有反馈项，而且使用有限系数实现，所以输出永远不会是无限的。

计算效率：位于单位圆附近的极点导致它们附近频率范围内增益增加。位于或者接近单位圆的零点导致附近频率范围内增益减小或者降到 0。对于 FIR 滤波器，首先设置增益为单位增益，然后通过零点的放置，可以有选择性地降低相应频率范围的增益。

线性相位：FIR 滤波器始终可以实现线性相位，而且 FIR 滤波器设计成线性相位也比较简单。实际应用中线性相位指标往往是选择 FIR 滤波器的主要因素。

量化与舍入误差：相比较 IIR 滤波器，误差问题对于 FIR 滤波器不明显。原因在于引入的任何误差都不会经过反馈而保留在系统当中。

VLSI 实现：FIR 滤波器的结构使得易于以 VLSI 结构实现，可由一系列的延迟器与乘法器设计构成。

设计难度：相比较 IIR 滤波器，FIR 滤波器设计起来稍显复杂。简单的 IIR 滤波器(如谐振器和陷波滤波器)可以手动设计实现。

习题

概念题

1. 为什么实际滤波器的相位曲线总是以负斜率开始？
2. 已知滤波器的脉冲响应，简述确定滤波器频率响应的方法。
3. 图 P5.3 给出了 FIR 滤波器的相位曲线。请问该滤波器位于单位圆上的零点的数目最少是多少？并证明你的答案。
4. 简述利用窗函数法设计 FIR 滤波器过程中需要注意的设计权衡。
5. 理想窗函数有哪些特征？
6. 简述利用窗函数法设计 FIR 滤波器时，为何窗函数频率响应的主瓣宽度是选择窗函数的重要考量？
7. 假设已知一个使用汉明窗设计的加窗 FIR 滤波器的传输函数(原始未加窗的滤波器已经丢失)。请详细描述如何使用布莱克曼窗来重新设计此滤波器。
8. 如果 FIR 滤波器窗函数在频域的主瓣非常宽，以此设计出的滤波器将会具有什么样的特征(从波动和过渡带宽度分析)？

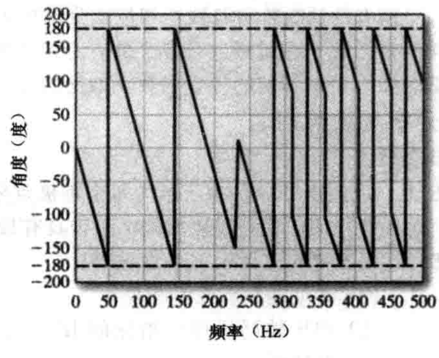


图 P5.3 一个 FIR 滤波器的相位曲线

9. 一个星期六的晚上，作为一个电子工程设计师的你无事可做、无处可去。偶然地进入了一个酒吧，并点了一杯葡萄酒，并向酒保解释你设计了一个数字滤波器，但是相位曲线在许多地方不连续。接着你咨询酒保是否可以解释这种情况或者提供一些帮助来解决这个神秘的问题。酒保给予了你两个解释，请猜猜它们是什么？
10. 许多 FIR 滤波器具有位于单位圆上的零点，这些零点导致相位曲线上的不连续。因此，频域内一个微小的改变导致了信号相位平移了接近 180°。请解释为什么这不能用来传递信息？
11. 假定已知如下 FIR 滤波器

$$H(z)=\frac{b_0z^M+b_1z^{M-1}+\cdots+b_M}{z^M}$$

(a) 如果将分子项都乘以 z，得到：

$$H(z)=\frac{b_0z^{M+1}+b_1z^M+\cdots+b_Mz}{z^M}$$

对于实际系统的实现，其意味着什么？

(b) 如果将分母项都乘以 z，于是可以得到：

$$H(z)=\frac{b_0z^M+b_1z^{M-1}+\cdots+b_M}{z^{M+1}}$$

对于实际系统的实现，其意味着什么？

12. 线性相位 FIR 滤波器具有一个特点，其滤波器系数不是对称的就是反对称的。一般乘法器比加法器所需的计算时间要长，线性相位 FIR 滤波器的系数对称特点对系统实现意味着什么？
13. 通过去除一些样本得到较低抽样率的信号以实现信号的抽取，但要求新的抽样率与原先的抽样率是整数倍关系。若新的抽样率与原先的抽样率不是整数倍关系，如何进行信号的抽取？例如，抽样率为 1000Hz 的正弦信号，如何通过抽取得到抽样率为 600Hz 的信号？忽略其中的混叠问题。
14. 图 P5.14 所示的是由 MATLAB 设计的低通滤波器。

    (a) 该滤波器位于单位圆上的零点数目最少是多少？解释你的答案。

    (b) 该滤波器最小的阶数是多少？解释你的答案。

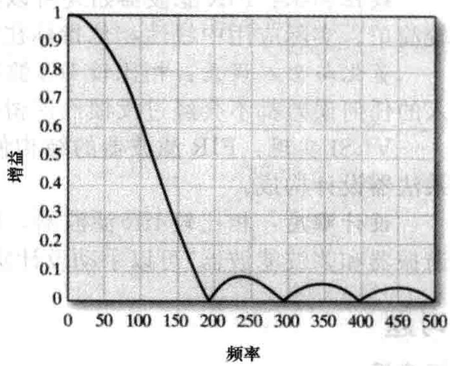


图 P5.14　低通滤波器

15. 如果 $H(z)$ 和 $G(z)$ 都是因果稳定的最小相位滤波器，那么 $H(z)/G(z)$ 滤波器是否是因果稳定的最小相位滤波器？解释你的答案。
16. 假设已知低通滤波器的截止频率 f_c，抽样频率 f_s。如果将抽样频率增加为原来的两倍并保持原来的截止频率，零极点之间变得更加接近、相互远离，还是其他情况？证明你的答案。
17. 假定使用帕克斯-麦克莱伦法设计的 FIR 滤波器截止频率为 5000Hz，抽样频率 44100Hz。如果滤波器实现时，设计者不小心将抽样频率设置成了 22050Hz，截止频率将会发生什么变化？为什么？
18. 微分器由于频率响应在 0Hz 处是 0，所以在 $Z=+1$ 处有一个零点。FIR 滤波器具有线性相位，为什么所有的偶数阶 FIR 线性相位微分器在 $f_s/2$ 处的增益是 0，而所有的奇数阶 FIR 线性相位微分器并不是？
19. 窗函数是在时域进行定义的，为什么要在频域使用卷积来解释它们的特性？
20. 如果已知 z 域的传输函数，如何确定 0Hz 与 $f_s/2$ 处的增益？

分析与设计问题

5.2 节

5.1　对于图 AP5.1 中所示的每个零极点分布图，确定相应的滤波器是 FIR 还是 IIR？是否稳定？是否因果？是否具有线性相位？是否具有最小相位？以及是什么类型(LP、HP、BP、BS)？

5.2　以下哪些适用于 FIR 滤波器？

(a) 有可能具有线性相位；

(b) 相比较同样设计指标的 IIR 滤波器，阶数较低；

(c) 在原点处有多重零点；

(d) 可能在 $Z=1$ 处有极点；

(e) 总是稳定的；

(f) 以上都成立；

(g) 以上都不成立。

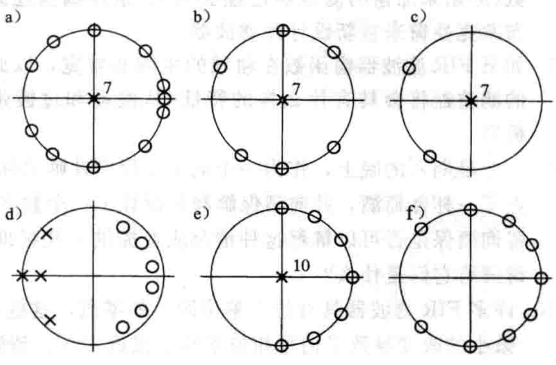


图 AP5.1　零极点特征

5.3　考虑简单的延迟线，该延迟线使输入信号延迟 9 个样本周期。

(a) 系统的脉冲响应是什么？

(b) 系统的传输函数是什么？

(c) 系统是否具有最小相位？解释你的答案。

5.4　设计满足以下指标的窗函数 FIR 滤波器，可以使用任意的设计方法，获得最小的滤波器阶数。
滤波器类型：低通
通带：0～150Hz
通带波动：0.03
阻带：200～500Hz
阻带波动：0.05
抽样频率：1000Hz

5.5　长度为 5 的三角窗函数的脉冲响应：

$$w_{\text{Tri}}[n]=\{0.333,0.667,1.0,0.667,0.333,0,0,\cdots\}$$

理想低通滤波器的脉冲响应如下：

$$h_{\text{I}}[n]=\{\cdots,0.2,0.3,0.6,0.8,\mathbf{1.0},0.8,0.6,0.3,0.2,\cdots\}$$

其中，粗体表示 $n=0$ 对应的项。求解此窗函数滤波器的传输函数。

5.6　利用 MATLAB 中的 firl 函数，分别画出使用长度为 6、9、12 和 15 的三角窗设计的 FIR 低通滤波器的幅频响应。抽样频率设置为 44.1kHz，截止频率为 8kHz。并解释不同长度滤波器带来的截断影响。

5.7　构造截止频率为 8kHz、抽样频率为 44.1kHz 的低通滤波器。使用汉明窗对此滤波器加窗，截断阶数为 12(长度为 13)。加窗滤波器的分子系数如表 AP5.7 所示，那么原始滤波器的系数是多少？增益$K=0.0022$。

表 AP5.7　加窗 FIR 滤波器

分子项	系数	分子项	系数
0	1.0	7	121.2170
1	−2.2210	8	41.5543
2	−10.8900	9	−7.0192
3	−7.0192	10	−10.8900
4	41.5543	11	−2.2210
5	121.2170	12	1.0
6	162.0577		

5.8　以下函数和图表示长度为 13 的指数窗。

$$h_e[n]=\begin{cases}k(e^{-|n|}-e^{-6}), & |n|\leqslant 6\\ 0, & \text{其他}\end{cases}$$

其中，设定 k 使窗函数最大值为 1。
(a) 利用 MATLAB 画出窗函数的频率响应；
(b) 利用该窗函数，结合理想低通滤波器构造 FIR 滤波器，要求截止频率为 150Hz，抽样频率为 1000Hz，画出频率响应。

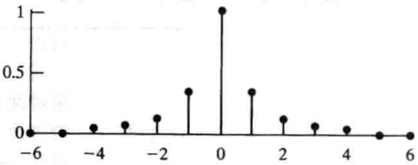


图 AP5.8　窗函数

5.9　使用理想滤波器的脉冲响应来构造加窗数字滤波器，该脉冲响应如下：

$$h[n]=\sum_{n=-\infty}^{\infty}\frac{1}{2^{|n|}}$$

根据此脉冲响应，所设计出的滤波器的传输函数为，

$$H(z)=\frac{z^4-3z^3+2z^2-3z+1}{z^4}$$

求解用来构造此滤波器的窗函数 $W(n)$。

5.10　带通滤波器可通过级联低通滤波器与高通滤波实现。使用 MATLAB 中的 firl 函数构造 20 阶的低通滤波器，抽样频率为 22050Hz，截止频率为 5500kHz，选择汉明窗进行加窗。接着再用 firl 函数构造 20 阶的高通滤波器，抽样频率为 22050Hz，截止频率为 3000Hz，同样选择汉明窗。画出每个滤波器的幅频图以及级联后的幅频图。

5.11 利用MATLAB中的firl函数构造一个12阶的FIR低通滤波器。抽样频率为1000Hz，截止频率为200Hz，选择矩形窗函数。绘制滤波器幅频图以及零极点图，并且使用汉明窗重复滤波器设计。从z域的零点分布角度，比较两个滤波器的幅频图。另外，解释为什么相比较于汉明窗构造的滤波器，矩形窗构造的滤波器具有较多位于单位圆上的零点。

5.12 利用MATLAB分别绘制长度为30的矩形窗、三角窗、汉明窗、汉宁窗以及布莱克曼窗。

5.13 由截止频率为3000Hz的理想低通滤波器设计5个FIR滤波器，抽样频率为20000Hz。滤波阶数均为30，但是使用不同的窗函数，依次是：矩形窗、巴特利特(bartlett)窗、汉明窗、汉宁窗和布莱克曼窗。试确定对于每个滤波器过渡带(增益介于0.1(10%)～0.9(90%)之间的频带)，并与图5.17所示的过渡带的主瓣宽度进行比较。

5.14 图AP5.14给出三个FIR滤波器，三者都由同一个理想滤波器使用傅里叶级数方法而构造，区别之处在于使用不同的窗函数。请回答以下问题：

(a) 从三个滤波器输出中，给出一个N，使得此三个滤波器的阶数都大于或等于此N，并给予相应的解释。

(b) 从三个滤波器输出(1、2和3)中，指出哪一个滤波器的窗函数的主瓣宽度最小。

(c) 如果这三个窗函数依次是矩形窗、汉明窗和布莱克曼窗，那么请指出每个滤波器对应的窗函数。

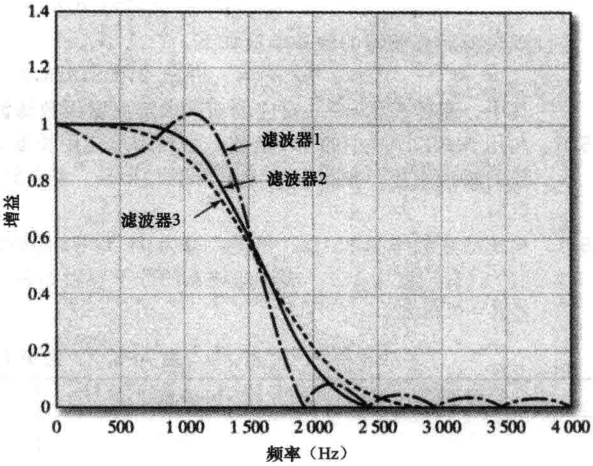


图 AP5.14 使用三个不同的窗函数构造的滤波器

5.15 FIR滤波器的阶跃响应如下：

$$s[n]=\{0,0.5,1,1.75,2,2,2,2,\cdots\}$$

求滤波器z域的传输函数。

5.16 设计加窗FIR滤波器，要求阶数不大于20，且满足以下指标：

通带	0～1400Hz
通带波动	±0.005
阻带	3200Hz～$f_s/2$
阻带波动	−50dB
阶数	≤20
抽样频率	11025Hz

5.17 设计加窗FIR滤波器，满足以下指标：

通带	0～6000Hz
阻带	8500Hz～$f_s/2$
通带波动	±0.01
阻带波动	−70dB
抽样频率	44100Hz

5.18 设计线性相位FIR滤波器，阻带为3000Hz～$f_s/2$，波动小于0.01，通带为0～1200Hz。可以使用矩形窗、三角窗、汉宁窗、汉明窗或者布莱克曼窗，抽样频率可选择11025Hz或者22050Hz。在满足阻带指标的情况下，最小阶数滤波器对应的通带波动是多少?

5.19 图AP5.19给出了汉明窗归一化幅频图。增益归一化至1，而且$f_s/2$同样归一化到1。如使用图AP5.19定义斜率为Δ_{gain}/Δ_f，设计MATLAB程序生成斜率v_s与汉明窗滤波器长度间的关系图，并证明斜率与滤波器长度之间是线性关系。(其他的窗函数也满足这种关系。)

5.20 汉明窗与汉宁窗的函数由如下公式定义：

$$W_H[n]=\begin{cases}\alpha+(1-\alpha)\cos\left(\dfrac{2\pi n}{L-1}\right), & |n|\leqslant(L-1)/2\\ 0, & \text{其他}\end{cases}$$

其中，对于汉明窗 $\alpha=0.54$，汉宁窗 $\alpha=0.50$。利用上述公式设置 $\alpha=0.58$，构造一个新的窗函数，并将此窗函数与图 5.17 所示的汉明窗进行比较，对比两者的主瓣宽度与波动百分比。

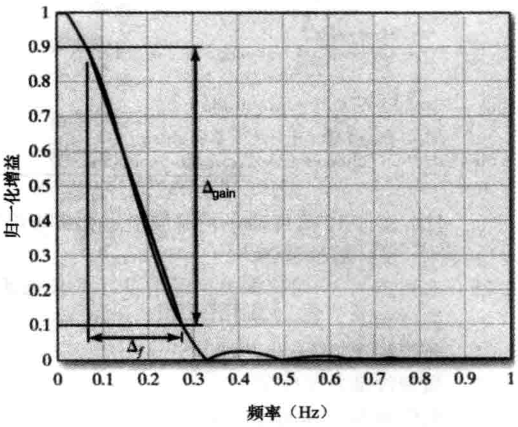


图 AP5.19　斜率定义为 Δ_{gain}/Δ_f

5.21　利用 MATLAB 分别绘制长度为 30 的矩形窗、三角窗、汉明窗、汉宁窗以及布莱克曼窗的脉冲响应。

5.22　在一次关于 FIR 滤波器的演讲中，教授问同学们如何设计带通滤波器。位于最后排的一名学生，在讲述频率分析和 z 变换的课堂上都在睡觉，站起来回答道：可以通过抽样一个频率对应带通滤波器中心频率的正弦波，并使用抽样值作为滤波器系数的方法实现带通滤波器。因此，如果中心频率是 1kHz、抽样频率是 12kHz，可通过正弦信号的抽样样本点作为传输函数分子多项式的系数 $b_0\sim b_{12}$，如图 AP5-22 所示。当此正弦信号通过该滤波器时，将对执行信号值平方操作，而其他正弦信号将变为较小的值。这样的论证有效吗？利用 MATLAB 设计一个 FIR 滤波器验证你的判断。

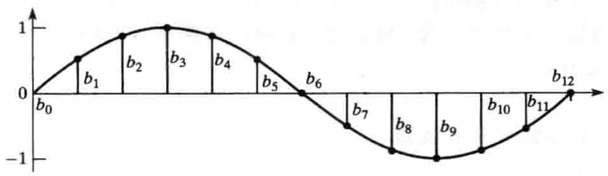


图 AP5.22　通过正弦信号的抽样是否可以构造带通滤波器

5.3 节

5.23　设计两个 FIR 低通滤波器。第一个使用凯瑟窗，第二个使用帕克斯-麦克莱伦方法。设计的滤波器需满足以下指标：

抽样频率	44kHz
通带	0～8kHz
通带波动	0.01
阻带	10～22kHz
阻带波动	0.02

绘制幅频图和相频图，验证设计的滤波器是否具有线性相位和满足设计指标。

5.24　利用下列 MATLAB 代码设计一个包含 3000 个元素的矢量 x，表示一个 5V 方波并伴随着±2V 噪声。高脉冲持续 200 个单位时间，低脉冲持续 400 个单位时间(占空比为 33.3%)。

```
fs = 1000;T = 1/fs;
t = 0:T:2999*T;
f0 = 5;
sqWv = 2.5*(square(2*pi*f0*t, 33.3)+1) + 4*(rand(1, 3000) - .5);
plot(t, sqWv);
axis([0 3 -2 8]);
```

(a) 选择凯瑟窗设计FIR低通滤波器满足以下指标，利用MATLAB中的filter函数仿真加噪方波通过此滤波器的结果，并绘制原始信号与滤波结果。

类型	FIR低通
通带	0Hz～$5f_0$
通带波动	0.1
阻带	$7f_0$～$f_s/2$
阻带波动	0.05
抽样频率	1000Hz

(b) 设计12阶滑动平均滤波器过滤含噪声信号。利用MATLAB中的filter函数进行仿真，验证所设计的滤波器。

5.25 选择道尔夫-切比雪夫窗函数设计FIR低通滤波器，要求满足以下指标：
抽样频率：f_s=44110Hz
通带频率：0～3000Hz
通带波动：±0.05
阻带频率：4600Hz～$f_s/2$
阻带波动：0.03

5.4节

5.26 构造一个19阶的频率抽样滤波器，要求具有对称响应。抽样频率为1000Hz，截止频率为200Hz。从截止频率为200Hz、通带增益为1、阻带增益为0的理想滤波器上选取初始样本点。由于滤波器长度为20，抽样率为1000Hz，因此频率样本点之间的间隔为1000/20=50Hz。编写MATLAB程序，改变200Hz处的样本点的幅值，使其从0.3变化到0.7，变化步长为0.01，当实际滤波器与理想滤波器之间的最大误差(绝对值)最小化时，确定此样本点的幅值。偶数长度对称频率抽样滤波器的公式$h(n)$见表5.2。

5.5节

5.27 设计三个满足以下指标的FIR滤波器：
抽样频率：22050Hz
通带频率：0～2500Hz
通带波动：0.01
阻带频率：3300Hz～$f_s/2$
阻带波动：0.01
分别使用凯瑟窗、道尔夫-切比雪夫窗和帕克斯-麦克莱伦算法。每种情况要求最小化滤波器阶数。

5.28 利用帕克斯-麦克莱伦算法设计满足以下指标的FIR滤波器。
抽样频率：22050Hz
通带频率：0～1500Hz
通带波动：0.01
阻带频率：2300Hz～$f_s/2$
阻带波动：0.02
要求最小化滤波器阶数。

5.29 下列MATLAB代码设计包含2500个元素的矢量x，其表示的是一个占空比为50%的1V方波。利用帕克斯-麦克莱伦方法设计FIR低通滤波器，方波可以通过此滤波器，方波的第3次、第5次谐波少于5%的幅度衰减，第9次和更高次的谐波被衰减到5%以下。利用MATLAB中的filter函数仿真x通过该滤波器的结果，并绘制原始方波与滤波后的结果。

```
fs = 44100;T = 1/fs;
pts = 2500;
f0 = 500;
terms = 18;
x = fouriersq(fs, pts, f0, terms);
%fouriersq is a function in Appendix D
```

5.30　下列 MATLAB 程序生成频率为 1000Hz 的正弦信号的 2000 个样本点，该正弦信号混入了两个其他的正弦信号，其中一个频率为 5000Hz，另一个频率为 400Hz。抽样频率为 16000Hz。

```
fs = 16000;T = 1/fs;
f = 1000;f1 = 5000;f2 = 400;
k = 0:1999;
xo = sin(2*pi*f*k*T);
n = .5*sin(2*pi*f1*k*T) + .25*sin(2*pi*f2*k*T);
x = xo + n;
```

(a) 设计 FIR 低通滤波器将 5000Hz 的正弦信号增益减小到 0.01 以下，1000Hz 的正弦信号增益为 1.0 ± 0.01。要求选用帕克斯-麦克莱伦算法并最小化滤波器阶数，即设计出最小阶数帕克斯-麦克莱伦 FIR 滤波器以满足设计指标。

(b) 设计 FIR 高通滤波器将 400Hz 的正弦信号增益减小到 0.01 以下，1000Hz 的正弦信号增益为 1.0 ± 0.01。要求选用帕克斯-麦克莱伦算法并最小化滤波器阶数。

(c) 利用 MATLAB 中的 filter 函数仿真设计的低通滤波器、高通滤波器以及它们级联的结果。

5.6 节

5.31　对于图 AP5.31 的每个零极点，增加最少数目的零点，使得产生的 FIR 滤波器具有线性相位。

5.32　针对图 AP5.32 所示系统，回答问题：

(a) 求解系统脉冲响应；

(b) 求解系统 z 域传输函数；

(c) 此系统是否是线性相位？并证明你的答案。

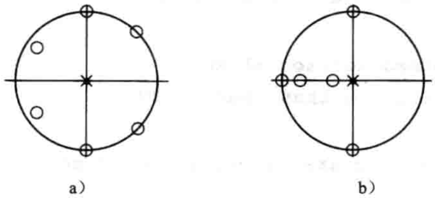


图 AP5.31　对每个图增加一些零点以获得线性相位

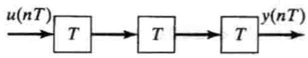


图 AP5.32　一个延迟线

5.33　下面的两个 FIR 滤波器，一个具有最小相位，另一个具有最大相位，分别指出对应的传输函数，并证明你的答案。

$$H_1(z)=\frac{z^2+1.2135z+0.5625}{z^2}\quad H_2(z)=\frac{z^2+2.1574z+1.7778}{z^2}$$

5.34　下表给出了抽样频率为 44100Hz，截止频率为 3000Hz 的 FIR 低通滤波器的部分零极点。添加最少数量的零点和极点，使得滤波器具有线性相位，而且依然是因果的 FIR 滤波器。

极点	零点
0	0.4982+j0.8670
0	0.4982−j0.8670
	−0.4276+j0.9039
	−0.351+j0.606

5.35　某 FIR 滤波器的传输函数 $H(z)=0.5\,\frac{(z+0.5)(z+2)}{z^4}$

(a) 0Hz 处的相移是多少？

(b) 修改传输函数，对其增加一个零点，使得 FIR 滤波器在 0Hz 处相移为 90°。

5.7 节

5.36　某滑动平均滤波器得到的输出 y_n 是其最近的 n 项输入之平均值。长度为 3 的滑动平均滤波器的传输函数为

$$H_{MA3}=\frac{z^2+z+1}{3z^2}$$

求解两个这样的滑动平均滤波器级联的脉冲响应。

5.37 如下所示为N阶希尔伯特变换器的相移曲线，根据该图形试确定N的值。抽样频率为1000Hz。

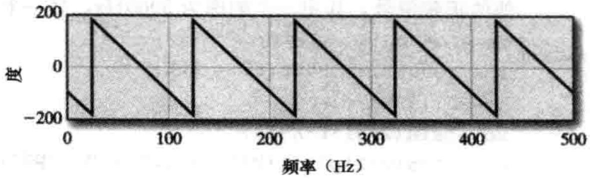


图 AP5.37 N阶希尔伯特变换器的相移曲线

5.38 利用MATLAB设计30阶希尔伯特变换器与15阶延迟器。抽样频率f_s为10000Hz。产生输入信号$x[n]=\cos(2\pi fnT)$，其中，$f=800$Hz，$T=1/f_s$，向量$x[n]$至少有2048个样本点。利用MATLAB中的filter函数分别确定希尔伯特变换器与延迟器的输出，并在同一图上绘制两个输出结果，通过比较确定两个信号之间的相位差。

5.39 电视中的水平与垂直同步信号常被混合传输或者组合使用同一根信号线。水平同步信号是频率为15750kHz的短脉冲信号，垂直同步信号为60Hz的长脉冲信号。为了将两个信号混合在一起，通过反转水平同步信号得到垂直同步信号，从而形成如图AP5-39所示的波形。如下MATLAB的m-file程序可以生成2000个单位时间长度的混合同步信号的仿真。将此混合信号通过微分器可分离出水平同步信号。设计离散微分器用于分离水平同步信号。使用firpm函数或者firls函数构造微分器，通过MATLAB中的filter函数生成仅含有水平同步信号的微分后的输出数据。

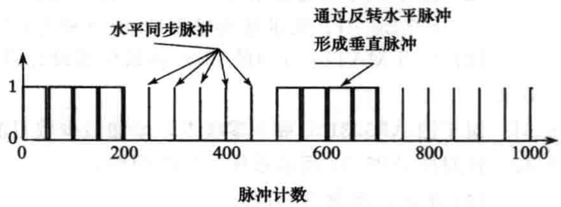


图 AP5.39 电视同步信号

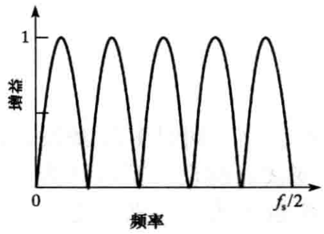
```
% Diff.m
% This m-file generates a simulated mixed horizontal and
% vertical synchronization signal similar to that used in TV.
i = 0;
while i < 2000                  % This loop generates horizontal synch
    if (mod(i, 50) == 0)        % pulses
        x(i+1) = 1;
        x(i+2) = 1;
        i = i + 2;
    else
        x(i+1) = 0;
        i = i + 1;
    end
end
i = 0;
while (i < 2000)                % This loop inverts selected
    if(mod(i, 500) == 0)        % horizontal pulses to simulate
      for j = i:i+200           % vertical synch pulses.
        if(x(j+1) == 0)
          x(j+1) = 1;
        else
          x(j+1) = 0;
        end
      end
      i = i + 200;
    end
    i = i + 1;
end
```

5.40 假定利用FIR设计方法设计了10阶的梳状滤波器，如图AP5.40所示。回答以下问题：

(a) 该滤波器在 0Hz 处的相移是多少？

(b) 0Hz～$f_s/2$ 的相位曲线上将会出现多少不连续点(如果有)？

(c) 原点处需要多少个极点，才能使得该滤波器是因果的？



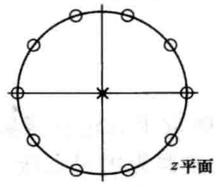


图 AP5.40 梳状滤波器

5.41 写出 5 阶滑动平均 FIR 滤波器的脉冲响应。

第6章

IIR 滤波器设计与分析

FIR 滤波器设计侧重于其传输函数的零点位置，同时 FIR 滤波器传输函数的极点都在原点处以使得 FIR 滤波器为因果系统。FIR 滤波器的设计可以认为是首先使所有频率的增益相同，然后再通过确定零点以调整指定频率范围的增益来实现。

IIR 滤波器含有反馈，使得其传输函数的极点不在原点位置上。极点会引起特定频率范围的增益增加，极点离单位圆越近，相应频率范围的增益增加越大。在 IIR 滤波器设计中，需要考虑如何设置其传输函数的零点或极点，以相应地减小或增加增益。综合考虑极点和零点位置，可以得到比 FIR 滤波器更高的计算效率和更窄的过渡带。

人们可能会疑问，既然一般而言 IIR 滤波器计算更高效，为什么还要设计 FIR 滤波器呢？主要有两个原因：滤波器的稳定性和线性相位。IIR 滤波器因为存在反馈可能会使系统不稳定，而 FIR 滤波器则不存在稳定性问题。反馈还会引起另一个与稳定间接相关的问题，即需要处理在反馈环中循环的噪声——这在 FIR 滤波器中也不会存在。正如我们所见，FIR 滤波器可以设计成线性相位滤波器。事实上，所有 FIR 滤波器设计技术所得到的滤波器都具有线性相位。而对于 IIR 滤波器来说，在 $0\sim f_s/2$ 频带上无法设计成线性相位，只能通过一些补偿技术在一定程度上逼近线性，相比较而言，FIR 滤波器更容易实现线性相位。

若能找到合适的映射函数将 s 平面映射到 z 平面，即可以通过已知的模拟滤波器来设计 IIR 数字滤波器。经典的巴特沃斯(Butterworth)、切比雪夫(Chebyshev)和椭圆(elliptic/Cauer)模拟滤波器都是很好的设计模板，可以通过使用诸如双线性变换这样的映射函数将它们转化成 IIR 数字滤波器。本章开始阶段使用这个技术来设计 IIR 数字滤波器。近年来不通过模拟滤波器也可以直接设计 IIR 数字滤波器，这种方式称为直接设计。在直接设计中，首先先设定初始的频率指标，再利用优化技术寻求合适的数字滤波器。本章后半部分将介绍如帕德(Pade)、普罗尼(Prony)、尤利-沃克(Yule-Walker)直接设计法，并以应用作为本章的结束。

6.1 利用双线性变换法设计 IIR 滤波器

一种设计 IIR 滤波器的方法是将连续时间滤波器(s 域传输函数)转化为数字滤波器(z 域传输函数)。一个连续时间低通滤波器的传输函数可表示为：

$$H(s)=\frac{Y(s)}{X(s)}=\frac{b_0s^m+b_1s^{m-1}+\cdots+b_m}{s^n+a_1s^{n-1}+\cdots+a_n},\quad n>m$$

上式中，分子分母同乘以 s^{-n}，交叉相乘，求解 y 可得

$$Y(s)=b_0s^{m-n}X(s)+\cdots+b_ms^{-n}X(s)-a_1s^{-1}Y(s)-\cdots-a_ns^{-n}Y(s)\tag{6.1}$$

第1章中曾提到，若 $y(nT)=\int_0^{nT}f(t)\mathrm{d}t$，则可用一个差分方程近似求解 y，差分方程的计算是基于梯形积分法，形式如下

$$y(nT)=y([n-1]T)+(T/2)[f(nT)+f([n-1]T)]$$

式(6.1)中，$Y(s)$值与 $X(s)$值都除以 s 的幂，对于拉普拉斯变换为 $G(s)$的任意函数 $g(t)$，$G(s)/s$ 的逆变换相应为 $\int g(t)\mathrm{d}t$。式(6.1)中，$s^{-1}Y(s)$项可使用差分方程项 $y(nT)=$

$y([n-1]T)+T*[y(nT)+y([n-1]T)]/2$ 进行替换。其相应的 z 变换为 $z^{-1}Y(z)+TY(z)*(1+z^{-1})/2$ 。使用如下替换也可得到相同的结论

$$Y(s)\rightarrow Y(z)$$

以及

$$s\rightarrow \frac{2}{T}\frac{z-1}{z+1}$$

或者

$$Y(z)=Y(s)\mid_{s=\frac{2}{T}\frac{z-1}{z+1}}(\text{双线性变换}) \tag{6.2}$$

式(6.2)即为所谓的双线性变换(Bilinear Transform，BLT)，其可以将传输函数从 s 域映射到 z 域。在通过转换模拟滤波器来设计数字滤波器的设计方法中，双线性变换扮演着重要的角色。

这里的双线性变换由梯形数值积分法得出。我们将会看到，其也可以仅由映射性质推导而来。

例 6-1 双线性变换的应用

如图 6.1 所示 RC 网络，利用双线性变换求解等效的数字滤波器。抽样频率 $f_s=1000\text{Hz}$, $R=1000\Omega$, $C=1\mu\text{F}$ 。

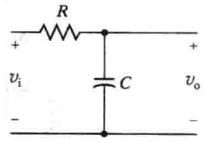


图 6.1　RC 电路。利用双线性变换求解等效数字滤波器

解：模拟滤波器的传输函数为

$$H(s)=\frac{1000}{s+1000}$$

利用双线性变换，可得到对应的数字滤波器的传输函数为

$$H(z)=\frac{1000}{\frac{2}{T}\frac{z-1}{z+1}+1000}=\frac{1000T(z+1)}{z(1000T+2)+(1000T-2)}$$

已知抽样频率为 1000Hz($T=0.001$)，整理上式得

$$H(z)=\frac{1}{3}\frac{(z+1)}{z-1/3}$$

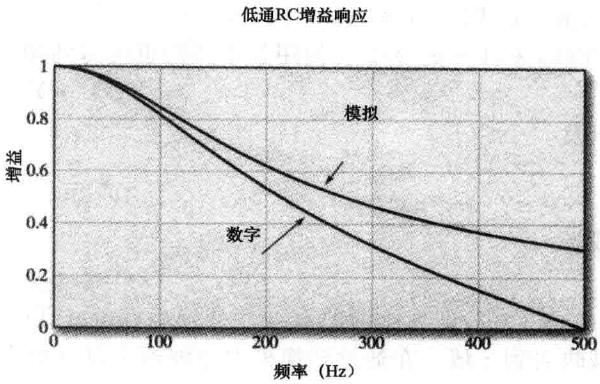
可以看到，s 域传输函数在 s 平面没有零点但有一个实极点，这使得 z 域传输函数在 z 平面 $z=-1$ 处有一零点，在 $z=1/3$ 处有一极点。一般地，当 s 域传输函数通过双线性变换被映射为 z 域传输函数时，若 s 域传输函数的分子为 M 阶，分母为 N 阶，则 z 域传输函数在 $z=-1$ 处有 $N-M$ 个零点。

数字滤波器及其对应模拟滤波器的幅频响应如图 6.2 所示。

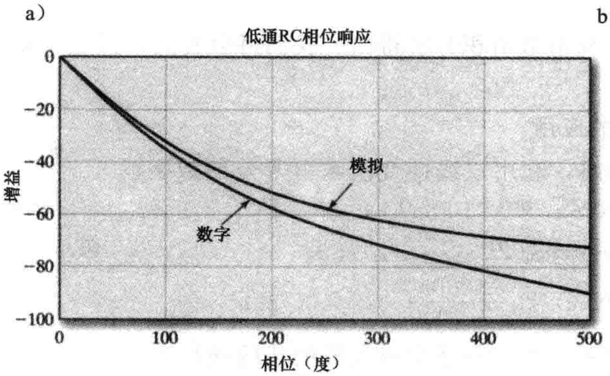
```
%RC.m
fs = 1000;T = 1/fs;
R = 1000;C = 0.000001;
Tau = R*C;
num = [1 1];
den = [1 -exp(-T/Tau)];
figure(1);clf;
[H f] = freqz(num, den, 1024, fs);
Habs = abs(H)/max(abs(H));
subplot(2, 1, 1);
plot(f, Habs);
subplot(2, 1, 2);
plot(f, angle(H)*180/pi);
```

```
numA = 1/Tau;denA = [1 1/Tau];
Ha = freqs(numA, denA, 2*pi*f);
subplot(2, 1, 1);hold on;
plot(f, abs(Ha), 'r');
title('Low pass RC Gain response');
xlabel('frequency in Hz');
ylabel('Gain');
subplot(2, 1, 2);hold on;
plot(f, angle(Ha)*180/pi, 'r');
title('Low pass RC Phase resp');
xlabel('frequency in Hz');
ylabel('Phase in degrees');
```

a)



b)



c)

图 6.2 模拟低通 RC 电路及其由双线性变换得到的数字滤波器幅频曲线及相频曲线

由例 6-1 可见，可以通过模拟滤波器作为模版，利用双线性变换设计数字滤波器。分析显示，幅频曲线和相频曲线都有一些失真，且频率越高失真越明显。为了进一步分析这种失真的原因，我们可以深入地剖析双线性变换将滤波器从 s 平面转变到 z 平面的过程。

求解式(6.2)可得

$$z=\frac{2+sT}{2-sT}$$

令 $s=\sigma+\mathrm{j}\omega$，$z=\alpha+\mathrm{j}\beta$。

将其代入上面的映射函数，可得

$$\alpha+\mathrm{j}\beta=\frac{(2+\sigma T)+\mathrm{j}\Omega}{(2-\sigma T)-\mathrm{j}\Omega},\quad 其中\ \Omega=\omega T$$

分子分母同乘以分母的共轭复数得

$$\alpha+\mathrm{j}\beta=\frac{-(\sigma T)^2-(\Omega)^2+4+\mathrm{j}(4\Omega)}{(\sigma T)^2-4\sigma T+(\Omega)^2+4}\tag{6.3}$$

若确定 $\mathrm{j}\omega$ 轴如何映射到 z 平面，令 $\sigma=0$。整理式(6.3)得

$$\alpha+\mathrm{j}\beta=\frac{4-(\Omega)^2+4\mathrm{j}\Omega}{4+(\Omega)^2}$$

求上式的幅值得

$$|\alpha+\mathrm{j}\beta|=\frac{[16-8(\Omega)^2+(\Omega)^4+16(\Omega)^2]^{\frac{1}{2}}}{[4+(\Omega)^2]}=1$$

由此可见，s 平面的 $\mathrm{j}\omega$ 轴映射到 z 平面为单位圆($\alpha^2+\beta^2=1$)。但 $\mathrm{j}\omega$ 轴为无限长，而单位圆的长度是有限的。表 6.1 列出了部分 $\mathrm{j}\omega$ 轴频率与单位圆的映射关系。该映射关系以图形表示如图 6.3 所示。由图可见，从 s 平面映射到 z 平面存在明显的频带失真。这种

失真称为"畸变"，z 平面的频率空间是"畸变的"。在 z 平面中，单位圆的上半部分类似机械欧姆表的刻度，最右边对应刻度的 0 欧姆，最左边对应刻度的无穷大欧姆。

表 6.1 该表显示了如何将频率通过双线性变换从 s 平面映射到 z 平面

s 平面 ω	z 平面双线性变换映射	s 平面 ω	z 平面双线性变换映射
0	+1	ω_S	$1\angle 144.7°$
$\omega_S/4$	$1\angle 76.3°$	$2\omega_S$	$1\angle 161.9°$
$\omega_S/2$	$1\angle 115.0°$	∞	−1

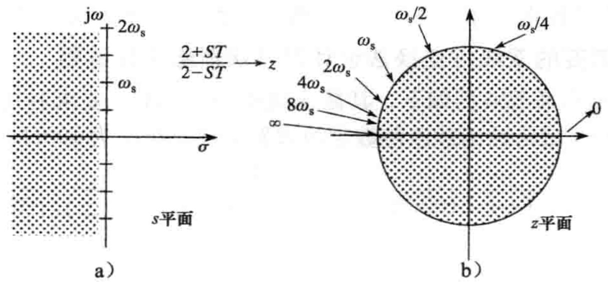


图 6.3 此图说明了通过双线性变换如何将 s 平面映射到 z 平面。jω 轴映射到单位圆，s 平面左半部分全部映射到 z 平面单位圆内部

因为已知 jω 轴映射到单位圆，所以有如下表示式，

$$s=\frac{2}{T}\frac{z-1}{z+1},\text{且 } \mathrm{j}\omega=\frac{2}{T}\frac{\mathrm{e}^{\mathrm{j}\Omega}-1}{\mathrm{e}^{\mathrm{j}\Omega}+1}$$

分子和分母同时提出因子 $\mathrm{e}^{\mathrm{j}\omega T/2}$ 可得

$$\mathrm{j}\omega=\frac{2}{T}\frac{\mathrm{e}^{\mathrm{j}\Omega/2}-\mathrm{e}^{-\mathrm{j}\Omega/2}}{\mathrm{e}^{\mathrm{j}\Omega/2}+\mathrm{e}^{-\mathrm{j}\Omega/2}}$$

根据欧拉公式，上式可化简为

$$\omega=\frac{2}{T}\frac{\sin(\Omega/2)}{\cos(\Omega/2)}$$

或

$$\text{预畸变公式}:\omega=\frac{2}{T}\tan(\Omega/2) \tag{6.4}$$

图 6.4 显示了 $f_S=1$ 时的预畸变方程曲线。虚线是一条参考直线。

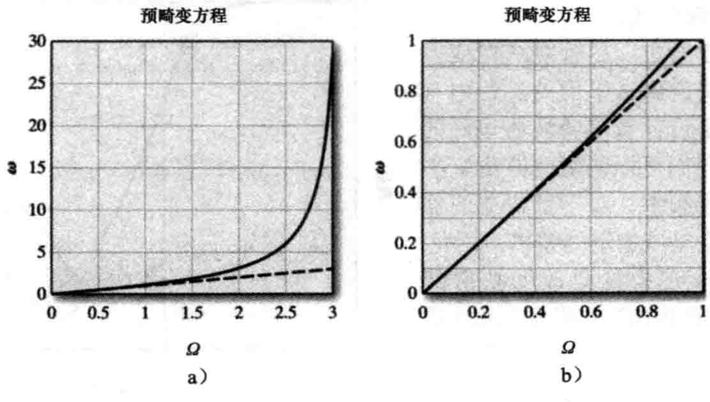


图 6.4 显示了 $f_S=1$ 时的预畸变方程曲线。虚线标明的是一条参考直线。可见当频率较低时，预畸变关系基本为线性

式(6.4)将 s 平面 jω 轴的频率与 z 平面单位圆的频率对应起来。该式给出了在 s 平面重新定位频率的方法，以便在应用双线性变换时，s 平面上的频率点能够映射到 z 平面上

适当的位置。例如，假设有在s平面的低通滤波器，其截止频率为250Hz。现要设计一个相同的数字滤波器，抽样频率为 $f_s=1000$Hz。不难看出，若直接应用双线性变换，已确定的截止频率将会略有变动。为了避免这一问题，可以根据预畸变方程将s平面的截止频率从250Hz改为新值。若要使z平面的截止频率为250Hz，可令 $\omega=2\pi\times250=500\pi$，从而由式(6.4)得到$s$平面预畸变后的截止频率

$$\omega_C=(2/T)\tan(500T/2)=2000\tan(0.5\pi/2)=2000\text{rad/s}=318.31\text{Hz}$$

318.31Hz即为s平面滤波器预畸变后得到的截止频率。再对预畸变的模拟滤波器应用双线性变换，可得到截止频率为250Hz的数字滤波器。下面通过例6-2详述此过程。

例 6-2 利用预畸变的双线性变换法设计巴特沃斯数字滤波器

已知模拟巴特沃斯低通滤波器如下，其截止频率为190Hz，要求利用双线性变换设计相应的数字滤波器，预畸变s域传输函数以使数字滤波器的截止频率为190Hz。抽样频率为1000Hz。

$$H(s)=\frac{K}{(s+1102.9\pm j456.85)(s+456.85\pm j1102.9)}$$

或

$$H(s)=\frac{K}{s^4+3119.6s^3+4.8658\times10^6s^2+4.4459\times10^9s+2.0311\times10^{12}}$$

解：首先对截止频率应用预畸变公式(6.4)，已知 $T=0.001$s。

$$\omega=(2/0.001)\tan(0.001(2\pi\cdot190)/2)=1359.2\text{rad/s}=216.32\text{Hz}$$

采用上式值作为巴特沃斯模拟滤波器的新截止频率。以预畸变的截止频率应用MATLAB生成一个新的巴特沃斯滤波器，传输函数如下

```
[num den]= butter(4,  2* pi* 216.32,  's');
```

$$H(s)=\frac{K}{(s+1255.7\pm j520.14)(s+520.14\pm j1255.7)}$$

或

$$H(s)=\frac{K}{s^4+3.5517s^3+6.3073s^2+6.5613\times10^6s+3.4128\times10^9}$$

对上式应用双线性变换，得到巴特沃斯数字滤波器的传输函数

$$H(z)=\frac{K_z(z+1)^4}{z^4-0.93911z^3+0.76607z^2-0.22937z+0.036038}$$

图6.5绘出了初始巴特沃斯滤波器及其数字滤波器的幅频曲线。

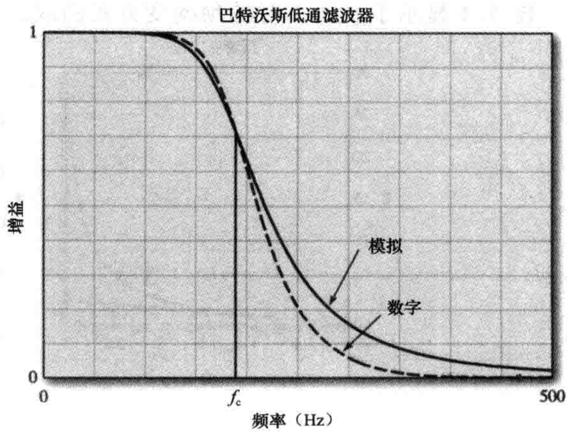


图6.5 4阶巴特沃斯模拟滤波器及其应用双线性变换法设计的数字滤波器 ◀

从例6-2可以看出，尽管双线性变换使频率从s平面映射到z平面的过程中有畸变，但它仍然是设计数字滤波器很好的工具。预畸变使我们可以明确地匹配z平面和s平面的指定频率；对于低通滤波器来说，截止频率常是该指定的频率。

另一种预畸变的方法是改变映射函数，而不需要生成新的"预畸变"滤波器作为中间步骤。由式(6.2)，双线性变换的方程如下

$$s\leftarrow\frac{2}{T}\cdot\frac{z-1}{z+1}$$

上式中，前面的项(2/T)是常数。因此可将上式一般化

$$s \leftarrow C \cdot \frac{z-1}{z+1} \tag{6.5}$$

这里 C 可以选择作为满足预畸变要求的常数。因为 $j\omega$ 轴映射到单位圆，所以上式可改写为

$$\omega = C \cdot \tan(\omega T/2)$$

对上式求 C 得

$$C = \omega \cdot \cot(\omega T/2)$$

或

$$C = 2\pi f_\omega \cot(\pi f_\Omega / f_S)$$

这里 f_S 是抽样频率，f_ω 是 s 域的频率，而 f_Ω 是 z 域的频率。

通过上式可以求出指定频率对应的 C 值，使得在此指定频率处，s 域与 z 域具有相同的幅频响应。称此频率为参考频率 f_{ref}，则有

$$f_{\text{ref}} = f_\omega = f_\Omega$$

以及

$$C = 2\pi f_{\text{ref}} \cot(\pi f_{\text{ref}} / f_S) \tag{6.6}$$

利用上式进行预畸变，可以先设计一个模拟滤波器，确定需要在 s 平面与 z 平面幅频响应保持一致的参考频率(如截止频率)，再由式(6.6)得到 C 值。将 C 值代入式(6.5)进行双线性变换，即得到预畸变效果的数字滤波器。

例 6-3 利用双线性变换设计巴特沃斯数字低通滤波器

已知巴特沃斯模拟低通滤波器，要求利用双线性变换设计相应的数字滤波器。利用改变映射函数的方法设计数字滤波器，使得其截止频率也是 190Hz。抽样频率为 1000Hz。

解：由式(6.6)，利用截止频率为 190Hz 作为 f_{ref}，求出 C 值。

$$C = 2\pi f_{\text{ref}} \cot(\pi f_{\text{ref}} / f_S) = 2\pi \cdot 190 \cdot \cot(190\pi/1000) = 1756.6$$

模拟低通滤波器的原传输函数为

$$H(s) = \frac{K}{s^4 + 3119.6s^3 + 4.8658 \times 10^6 s^2 + 4.4459 \times 10^9 s + 2.0311 \times 10^{12}}$$

应用式(6.5)进行双线性变换，则得出数字滤波器的传输函数为

$$H(z) = \frac{K_z (z+1)^4}{z^4 - 0.93904z^3 + 0.76602z^2 - 0.22934z + 0.036029}$$

该传输函数与例 6-2 通过预畸变方程得到的传输函数相同。注意，s 域传输函数的分母比分子高 4 阶，因而 z 域传输函数在 $z=-1$ 处有 4 个零点。 ◀

利用双线性变换设计数字滤波器的规则总结如下。

双线性变换设计规则

1. 首先在 s 域内确定满足设计指标的模拟滤波器的传输函数。

2. 确定数字滤波器需要保持的频率点。典型的频率点如低通滤波器的截止频率，而对于带通和带阻滤波器，则有多个频带边界频率。

3. 若要保持多个频率，则应利用预畸变方程(6.4)求解每个频率对应的预畸变值，且根据这些畸变值重构模拟滤波器。若有且只有一个频率需要保持，则可直接利用式(6.6)确定双线性变换的常数。

4. 对模拟滤波器应用双线性变换得到 z 平面的传输函数。在设计时可以不考虑增益常量，而是在设计完成后规一化滤波器的增益。

MATLAB 提供了能够利用双线性变换自动设计数字滤波器的函数，因此不需要像刚才几个例子那样做繁琐的计算。MATLAB 函数 bilinear 可将 s 域传输函数通过双线性变换产生 z 域的滤波器。传输函数以多项式或零极点形式表示，可选参数允许用户指定预畸变的参考频率。

如果基于诸如巴特沃斯、切比雪夫、椭圆等经典模拟滤波器，MATLAB在使用双线性变换产生对应的数字滤波器时，将会略去整个转换过程。

例 6-4 设计满足给定指标的巴特沃斯数字滤波器

利用MATLAB设计巴特沃斯数字滤波器，需要满足如下指标：

抽样频率	44100Hz
通带频率范围	0～4000Hz
通带增益	1
通带波动	0.01
阻带频率范围	6000～$f_s/2$
阻带增益	0
阻带波动	0.03

如下MATLAB代码得到满足给定指标的滤波器，其幅频响应如图6.6所示。

```
%Creates a Butterworth filter such that
%fs = 44,100 Hz
%pass band 0 - 4000 Hz; stopband 6000 Hz to fs/2
%pass band ripple = .01; stopband ripple = .03
fs = 44100;
fs2 = fs/2;
fpass = 4000; Rp = .01; RpDB = -20*log10(1-Rp);
fstop = 6000; Rs = .03; RsDB = -20*log10(Rs);
[N fc] = buttord(fpass/fs2, fstop/fs2, RpDB, RsDB);
[num den] = butter(N, fc);
[H f] = freqz(num, den, 1024, fs);
figure(1);clf;
subplot(2, 1, 1);
plot(f, abs(H));
title('Low-pass Butterworth filter');
axis([0 fs/2 0 1.2]);
xlabel('frequency in Hertz');
ylabel('gain');
subplot(2, 1, 2);
plot(f, 180*angle(H)/pi);
axis([0 fs/2 -180 180]);
xlabel('frequency in Hertz');
ylabel('Phase in degrees');
```

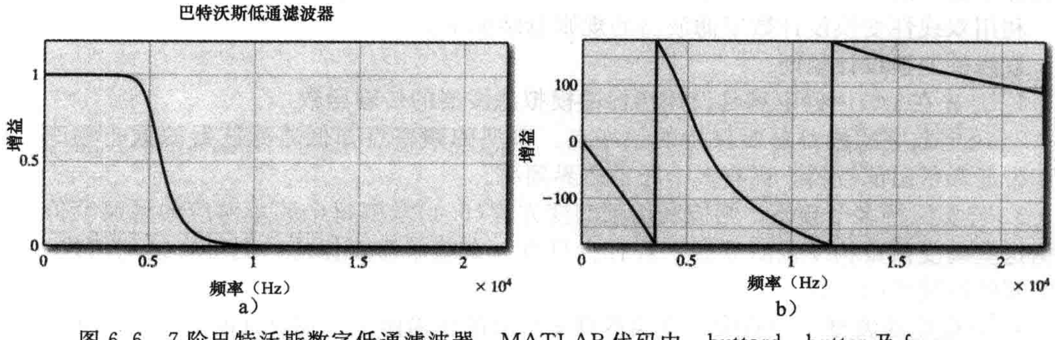


图 6.6 7阶巴特沃斯数字低通滤波器。MATLAB代码中，buttord、butter及freqz函数的语法参考附录D

6.2 IIR滤波器的稳定性

对于线性系统，若输入信号是0，则稳定输出信号也应是0，或者趋向0。对一个稳定

的数字滤波器输入脉冲信号，其响应经过一段时间(也可能是很长时间)后趋向 0。数字滤波器的传输函数的数学表示为：

$$H(z)=k\frac{b_0z^M+b_1z^{M-1}+\cdots+b_{M-1}z+b_M}{z^N+a_1z^{N-1}+\cdots+a_{N-1}z+a_N}$$

式中，k 是增益常量，在因果系统中 $M\leqslant N$。

假设 $H(z)$的分母有不重复根，可用部分分式法将传输函数展开成一阶项之和的形式。展开式形式如下

$$H(z)=k\left(k_0+\frac{k_1z}{z-p_1}+\frac{k_2z}{z-p_2}+\cdots+\frac{k_Nz}{z-p_N}\right)$$

式中，p_i 为 $H(z)$的极点，k_i 为常数。若 $M<N$，则 $k_0=0$。

展开式中的每个一阶项对应的 z 反变换如下

$$\frac{k_iz}{z-p_i}\leftrightarrow k_ip_i^n$$

通过该项考虑系统稳定性时，分以下三种情况：

1. 若极点值的模小于 1，即 $|p_i|<1$

此时，p_i^n 的值随 n 的增大而趋于 0，称系统稳定。

2. $|p_i|=1$

若极点值为 1，p_i^n 值既不趋于 0，也不会趋于无限大，称系统临界稳定。对于极点值的模为 1 的滤波器，系统不需要输入信号来产生输出信号，任何噪声的干扰(包括舍入噪声)或非零初始状态都会产生输出信号。有输出信号而没有输入信号的系统称为振荡器。滤波器的振荡是随机的。

3. $|p_i|>1$

若极点值的模大于 1，p_i^n 将会无限增大，这时系统不稳定。

极点的值可以是复数，我们得到这样的结论，若数字滤波器传输函数存在有位于 z 平面单位圆外的极点，则该滤波器是不稳定的。

例 6-5　转换传输函数为差分方程

考虑某一阶系统，其 z 域传输函数为 $H(z)=\dfrac{Y(z)}{U(z)}=k\dfrac{z-z_1}{z-p_1}$，$z_1$ 是传输函数的实数零点，p_1 是传输函数的实数极点，k 是增益常量。可以将传输函数转换为对应的差分方程，可以先将其写成 z 的负幂形式，交差相乘，用时移特性得到 z 反变换。得到的方程式如下：

$$y(n)=ku(n)-kz_1u(n-1)+p_1y(n-1)$$

取 $z_1=1$，$k=1$，画出 $p_1=0$，0.3，0.8，0.95，1.0，1.2 时系统的脉冲响应。

解：

```
T = 1;                                          % Sample period
r = [0 .3 .8 .95 1.0 1.2];
num = [1 1];                                    % numerator vector
for i = 1:6
  den = [1 -r(i)];                              % denominator vector
  [h nT] = impz(num, den, 40);
  figure(i);clf;
  stem(nT, h, 'MarkerEdgeColor', 'w');          % stem plots the result
end
```

如图 6.7 所示，脉冲响应或阶跃响应最易反映系统的不稳定性。典型的幅频响应无法反映出单位圆外的极点。如果一个系统是不稳定的，但它是线性的，当输入正弦信号时，则会输出一个同频率的正弦信号。

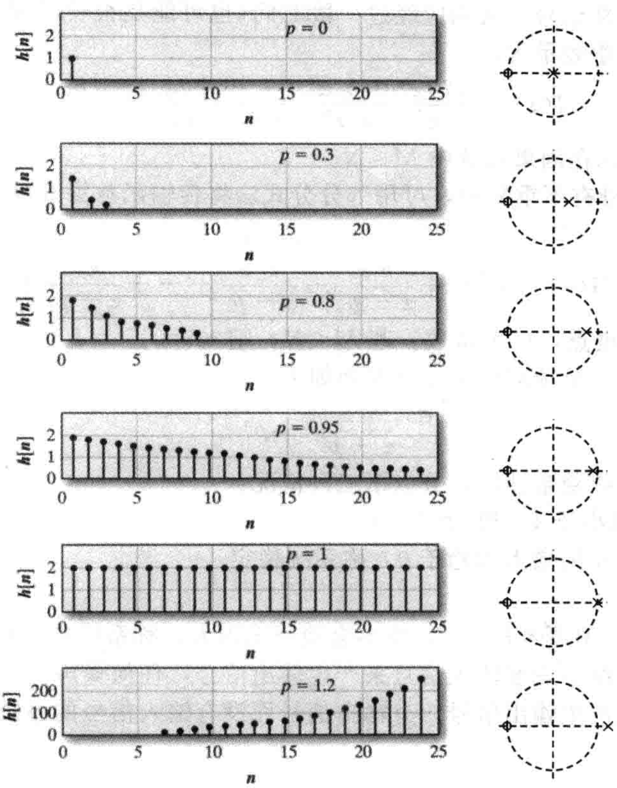


图 6.7　一阶系统的脉冲响应，零点在 $z=-1$ 处，极点位置从 0 变化到 1.2

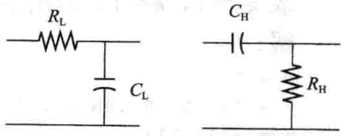
6.3　频率变换

到目前为止，我们分析的对象一直是低通滤波器。这是因为其他类型的滤波器如带通、带阻和高通滤波器等，都可以通过重新映射低通滤波器的频率轴来生成。此过程与前面双线性变换应用映射函数非常相似。

设计任意模拟滤波器常从设计一个归一化的模拟低通滤波器开始。归一化是指对于模拟滤波器选 $\omega_c=1$ 作为截止频率，对于数字滤波器选 $f_s=2\pi$ 作为抽样频率。若要得到非归一化的模拟滤波器，则需要用 s/ω_c 替换 $s=j\omega$。这种频率轴上的重新映射称为低通到低通变换。其作用是生成不同截止频率的低通滤波器。修改类似的重新映射方程，可由低通滤波器生成其他类型的模拟滤波器。

为进一步说明这个问题，考虑图 6.8 的两个 RC 电路。图 6.8a 为低通 RC 电路，图 6.8b 为高通 RC 电路。这两个电路的传输函数分别如下：

$$H_L(s)=\frac{1/(R_L C_L)}{s+1/(R_L C_L)} \tag{6.7}$$



a）低通RC电路　b）高通RC电路

图 6.8

以及

$$H_H(s)=\frac{s}{s+1/(R_H C_H)} \tag{6.8}$$

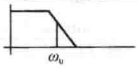
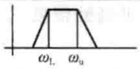
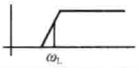
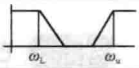
式(6.7)中，用 λ/s 替换 s。则新的重映射方程为

$$H_R(s)=H_L(s)\big|_{s\leftarrow\lambda/s}=\frac{s}{s+\lambda R_L C_L} \tag{6.9}$$

式(6.9)与式(6.8)有相同的形式。因此可知，通过选择合适的 λ 值，即可利用 $s \leftarrow \lambda/s$ 作为频率变换方程实现低通到高通的转换。

通过一定的代数运算，就可求解其他类型滤波器的频率变换方程。参照表 6.2。

表 6.2　模拟滤波器的频率变换。模拟低通滤波器的截止频率为 ω_c，通带边界频率为 ω_p。新的低通截止频率为 ω_u，高通截止频率为 ω_L

变换	映射函数	新滤波器	变换	映射函数	新滤波器
低通到低通	$s \leftarrow \omega_c s/\omega_u$		低通到带通	$s \leftarrow \omega_p \dfrac{s^2+\omega_L\omega_u}{s(\omega_u-\omega_L)}$	
低通到高通	$s \leftarrow \omega_c\omega_L/s$		低通到带阻	$s \leftarrow \omega_p \dfrac{s(\omega_u-\omega_L)}{s^2+\omega_u\omega_L}$	

MATLAB 正是应用这些频率变换以及通过双线性变换来实现设计各种数字滤波器。假如我们利用 MATLAB 设计切比雪夫数字带通滤波器，MATLAB 首先设计满足最坏情形波动指标的切比雪夫模拟低通滤波器。该模拟低通滤波器按照频带边界频率归一化。利用频率变换方程将模拟低通滤波器转换为模拟带通滤波器。需要注意的是，该过程会使低通滤波器的阶数加倍。然后对模拟带通滤波器进行预畸变，通过双线性变换即可生成切比雪夫数字带通滤波器。

综上所述，设计模拟高通、带通或带阻滤波器的过程如下。

1. 利用预畸变方程(6.4)对需要保持的频率进行预畸变——通常这些频率是频带的边界频率。
2. 利用表 6.2 所示的转换函数，反向代换求得低通滤波器对应的频带边界频率位置和波动。
3. 根据低通滤波器指标确定模拟滤波器的阶数。
4. 求出模拟低通滤波器的传输函数。
5. 应用表 6.2 的转换方程得到 s 平面内相应的模拟滤波器。
6. 应用双线性变换法得到相应的数字滤波器。

前面讨论的是通过模拟域频率变换的方法构造 IIR 数字滤波器。频率变换也可以在 z 平面进行。这时设计过程会有一些变动，如图 6.9 所示。在进行 z 域频率变换时，需要在 s 域对归一化模拟低通滤波器进行双线性变换，得到 z 域的归一化数字低通滤波器，再在 z 域对其进行频率变换得到最终的数字滤波器。可以看到，频率变换在 s 域或 z 域进行并不影响最终结果。表 6.3 列出了在 z 域的频率变换。

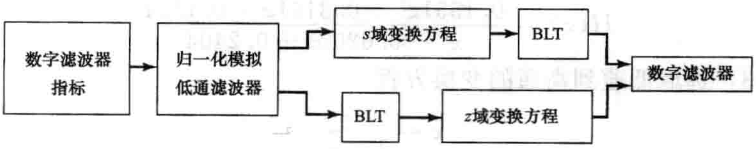
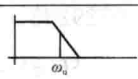
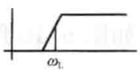
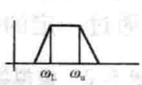
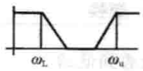


图 6.9　数字滤波器的设计过程

表 6.3　数字频率变换。待变换的低通原型滤波器的截止频率为 ω_c、ω_u 和 ω_L 分别表示新滤波器的上、下截止频率

变换	映射函数	常数	新滤波器
低通到低通	$z \leftarrow \dfrac{z-\alpha}{1-\alpha z}$	$\alpha = \dfrac{\sin[(\omega_c-\omega_u)T/2]}{\sin[(\omega_c+\omega_u)T/2]}$	
低通到高通	$z \leftarrow \dfrac{\alpha-z}{1-\alpha z}$	$\alpha = \dfrac{\cos[(\omega_L+\omega_c)T/2]}{\cos[(\omega_L-\omega_c)T/2]}$	

（续）

变换	映射函数	常数	新滤波器
低通到带通	$z \leftarrow \dfrac{-z^2+\dfrac{2\alpha k}{k+1}z-\dfrac{k-1}{k+1}}{\dfrac{k-1}{k+1}z^2-\dfrac{2\alpha kz}{k+1}+1}$	$\alpha=\dfrac{\cos[(\omega_u+\omega_L)T/2]}{\cos[(\omega_u-\omega_L)T/2]}$ $k=\dfrac{\tan(\omega_c T/2)}{\tan[(\omega_u-\omega_L)T/2]}$	
低通到带阻	$z \leftarrow \dfrac{z^2-\dfrac{2\alpha}{k+1}z+\dfrac{1-k}{1+k}}{\dfrac{1-k}{1+k}z^2-\dfrac{2\alpha z}{k+1}+1}$	$\alpha=\dfrac{\cos[(\omega_u+\omega_L)T/2]}{\cos[(\omega_u-\omega_L)T/2]}$ $k=\dfrac{\tan(\omega_c T/2)}{\cot[(\omega_u-\omega_L)T/2]}$	

例 6-6　巴特沃斯滤波器低通到高通的变换

利用MATLAB设计一个归一化的二阶巴特沃斯滤波器，使 $f_s/2=1$，最大通带增益为1，截止频率为1/3Hz。应用表6.3的低通到高通转换方程，将此滤波器转换为截止频率是5/8Hz的二阶高通滤波器。

解： 原型滤波器的MATLAB代码如下：

```
%NormalButter.m
%Creates a 2nd order normalized Butterworth filter with a cutoff
% frequency of 1/3.
fs = 2;
fc = 1/3;
N = 2;
[num den] = butter(N, fc);
[H f] = freqz(num, den, 1024, fs);
figure(1);clf;
plot(f, abs(H));
title('Second Order normalized Butterworth filter');
xlabel('frequency in Hz');
ylabel('Gain');
for i=1:3
    fprintf(1, '%2.12f\n', num(i));
end
for i=1:3
    fprintf(1, '%2.12f\n', den(i));
end
```

由以上代码可以得到分子和分母的系数，从而得到数字低通滤波器的传输函数：

$$H(z)=\frac{0.1551z^2+0.3101z+0.1551}{z^2-0.6202z+0.2404}$$

参考表6.3，选取低通到高通的变换方程

$$z \leftarrow \frac{\alpha - z}{1-\alpha z}$$

以及

$$\alpha=\frac{\cos[(\omega_L+\omega_c)T/2]}{\cos[(\omega_L-\omega_c)T/2]}$$

在此滤波器中，$\omega_c=2\pi(1/3)$，$\omega_L=2\pi(5/8)$，由 $f_s=2$Hz可知 $T=0.5$s，因此可求得 $\alpha=0.0729235$。将代换公式代入 $H(z)$ 得

$$H(z)=\frac{0.1785z^2-0.3570z+0.1785}{0.9561z^2+0.4426z+0.2004}=\frac{0.1867z^2-0.3734z+0.1867}{z^2+0.4629z+0.2097}$$

结果如图6.10所示。

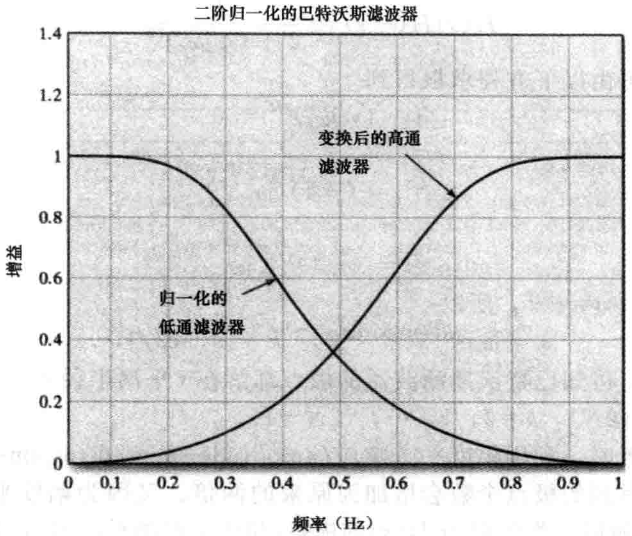


图 6.10 归一化的二阶巴特沃斯低通滤波器以及利用变换方程生成的高通滤波器

6.4 经典 IIR 滤波器

虽然没有“标准的”IIR 滤波器，但当考虑经典 IIR 滤波器设计时，一般都会想到巴特沃斯、切比雪夫及椭圆滤波器。因此，在数字领域也主要分析这三种滤波器。

从整体来看，这三种滤波器为设计者提供了权衡的选择，就像为 FIR 滤波器选择窗函数那样。巴特沃斯滤波器的通带平滑，阻带单调，过渡带在三种滤波器中最宽。椭圆滤波器在通带和阻带上都有波动，但其过渡带最窄。而切比雪夫Ⅰ型滤波器在通带中有波动，阻带单调，过渡带的宽度介于巴特沃斯滤波器和椭圆滤波器之间。

附录 A 介绍了模拟巴特沃斯、切比雪夫及椭圆滤波器。

6.4.1 巴特沃斯滤波器

1930 年，斯蒂芬·巴特沃斯(Stephen Butterworth)首次以一系列 RLC 电路描述了巴特沃斯滤波器，其频率响应函数如下

$$H(\omega)=\frac{1}{\sqrt{1+(\omega/\omega_c)^{2N}}} \tag{6.10}$$

图 6.11 给出了上式当阶数 $N=4$ 时的频率响应曲线。可以看到当频率为 0 时，其幅度为 1，而在 f_c 处，即半功率点(half-power point)处的幅度为 $\sqrt{2}/2$。f_c 称为截止(cutoff)频率(因为功率减半对应的衰减为 3dB，因此，常称 f_c 为 3dB 截止频率)。阻带是单调递减的，这意味着其导数总为负值。通带具有最大平坦特性且幅值总不超过 1。

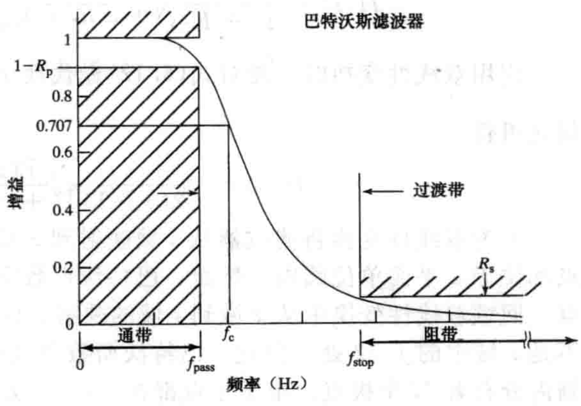


图 6.11 一个巴特沃斯低通滤波器，通带平坦，阻带单调，截止频率定义在增益为 $\sqrt{2}/2$ 的频率处

式(6.10)表示了巴特沃斯滤波器的幅频特性。模拟滤波器的传输函数以 s 表示，对于频率响应方程，取 $s=\mathrm{j}\omega$。式(6.10)可以写为

$$H(s)H(-s)=\frac{1}{1+(s/\mathrm{j}\omega_c)^{2N}} \tag{6.11}$$

式(6.11)中的极点可由以下方程求根得到

$$1+(s/\mathrm{j}\omega_c)^{2N}=0$$

或

$$s^{2N}=(-1)(\mathrm{j}\omega_c)^{2N}$$

及

$$s=(-1)^{1/2N}\mathrm{j}\omega_c$$

又因为$-1=\mathrm{e}^{\mathrm{j}(\pi+2k\pi)}$，$\mathrm{j}=\mathrm{e}^{\mathrm{j}\pi/2}$，所以

$$s=\omega_c\mathrm{e}^{\mathrm{j}\pi(1+N+2k)/(2N)},\quad k=0,1,2,3,\cdots$$

因为e^{j}项的模为1，可知巴特沃斯滤波器的极点都落在s平面半径为ω_c的圆上。极点的角度为$\pi(1+N+2k)/(2N)$，$k=0$，1，…，$(2N-1)$。

在求解极点位置时，利用幅度平方响应(amplitude-squared response)可以避免计算平方根。这意味着其得到的极点个数会增加为原来的两倍。又因为幅度平方响应由$H(s)$与$H(-s)$相乘得到，所以一半的根为$H(s)$的极点(位于s左半平面内)，另一半属于$H(-s)$的极点(位于s右半平面内)。

例 6-7　设计一个四阶巴特沃斯滤波器

设计一个截止频率为55Hz的巴特沃斯模拟滤波器。

解： 由$N=4$及k由0变化至7可得极点的角度分别为$5\pi/8$，$7\pi/8$，$9\pi/8$，…，$19\pi/8$。极点位置如图6.12所示。

极点项相乘即得到模拟滤波器的传输函数如下

$$H(s)=\frac{1}{s^4+903.03s^3+4.0773\times10^5s^2+1.0784\times10^8s+1.4262\times10^{10}}$$

若要得到相应的巴特沃斯数字滤波器，需对模拟滤波器的传输函数进行双线性变换。而在进行双线性变换之前，应对截止频率作预畸变，以使巴特沃斯数字滤波器更好地逼近相应的模拟滤波器。

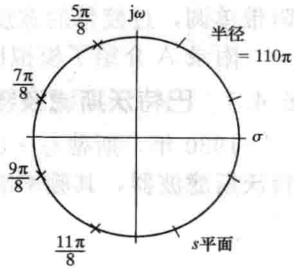


图 6.12　4阶巴特沃斯滤波器的极点分布

巴特沃斯模拟滤波器在s平面是“全极点”滤波器。它没有零点，且仅在$\omega=\infty$时幅值趋近于零。一般地，巴特沃斯模拟滤波器有如下形式

$$H(s)=\frac{1}{s^N+K_1s^{N-1}+\cdots+K_N} \tag{6.12}$$

应用双线性变换时，先对式(6.12)做代换$s\leftarrow\frac{2}{T}\frac{z-1}{z+1}$，再简化可得

$$H(s)=\frac{[T(z+1)]^N}{[2(z-1)]^N+\cdots+K_N[T(z+1)]^N} \tag{6.13}$$

因为双线性变换将滤波器从s域映射到z域时稳定性保持不变，所以式(6.13)中的极点都位于z平面单位圆内。然而，巴特沃斯数字低通滤波器在z平面$z=-1$处有N个零点。回顾双线性变换中从z域到s域的映射，注意，这些零点对应于s域中的无穷远处而不是z域中的$f_s/2$处。因此，巴特沃斯数字低通滤波器在$z=-1$处有N个零点，在单位圆内分布着N个极点。由于零点都在$z=-1$处，因此其传输函数的分子多项式系数具有对称性，在计算对应的差分方程时，可提高计算效率。

在s域设计巴特沃斯模拟滤波器，然后进行预畸变，再应用双线性变换得到数字滤波

器，此繁琐设计过程可由 MATLAB 来实现。

例 6-8 设计满足指标的巴特沃斯低通滤波器

利用 MATLAB 设计巴特沃斯数字低通滤波器，指标如下：

抽样频率＝11025Hz

截止频率＝2500Hz

阶数＝6

解：所求滤波器代码如下：

```
fs = 11025;    %sample frequency
fc = 2500;     %cutoff frequency
N = 6;         %order
[num den] = butter(N, fc/(fs/2)); %cutoff frequency is normalized
[H f] = freqz(num, den, 1024, fs);
plot(f, abs(H));
```

结果如图 6.13 所示。

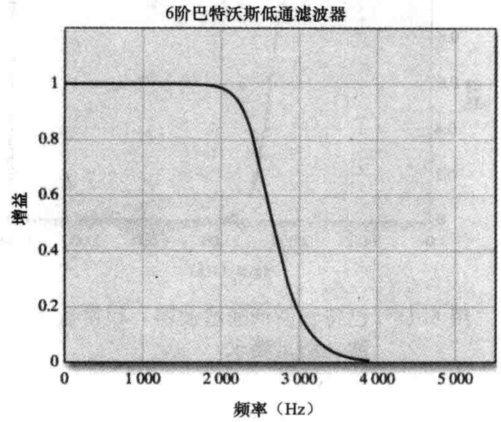


图 6.13 6 阶巴特沃斯低通滤波器，截止频率为 2500Hz，抽样频率为 11025Hz

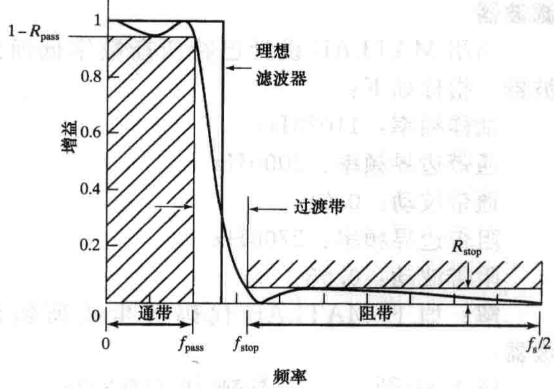


图 6.14 IIR 滤波器参数定义

在设计巴特沃斯滤波器时，往往不会给出滤波器的阶数与截止频率，更多的时候给出的参数是通带和阻带的边界频率以及通带和阻带的波动，如图 6.14 所示。因为需要知道截止频率来确定巴特沃斯多项式，所以可通过频带边界频率和波动信息及式(6.11)来确定截止频率和所需的阶数。

为计算巴特沃斯滤波器的阶数，定义如下符号：

f_{pass}——通带边界频率，单位是 Hz

f_{stop}——阻带边界频率，单位是 Hz

R_{pass}——通带波动

R_{stop}——阻带波动

R_{pass}和 R_{stop}是增益的比值且没有单位，也可转换为分贝。

通过这些定义可知，幅频响应在频率 f_{stop} 处的值为 R_{stop}，同样可知，在频率 f_{pass} 处的值为 $1-R_{pass}$。因此，由式(6.10)可列出含两个未知量的方程。

$$R_{stop}^2=\frac{1}{1+(f_{stop}/f_c)^{2N}} \tag{6.14}$$

$$(1-R_{pass})^2=\frac{1}{1+(f_{pass}/f_c)^{2N}} \tag{6.15}$$

应用代数方法即可解出方程中的 N 和 f_c，即

$$N = \frac{\ln(K_s/K_p)}{2\ln(f_{stop}/f_{pass})} \tag{6.16}$$

以及

$$\ln(f_c) = \frac{\ln(K_s)\ln(f_{pass}) - \ln(K_p)\ln(f_{stop})}{\ln(K_s/K_p)} \tag{6.17}$$

其中

$$K_s = (1/R_{stop}^2) - 1$$

$$K_p = \frac{1}{(1-R_{pass})^2} - 1$$

值得注意的是，若要利用双线性变换将模拟滤波器转换为数字滤波器，则频率 f_{pass} 及 f_{stop} 应是预畸变后的边界频率。

MATLAB 的 buttord 函数可自动完成式(6.16)及式(6.17)的计算。

例 6-9 设计满足指标的巴特沃斯数字低通滤波器

利用 MATLAB 设计巴特沃斯数字低通滤波器，指标如下：

抽样频率：11025Hz

通带边界频率：2000Hz

通带波动：0.01

阻带边界频率：2700Hz

阻带波动：0.05

解：如下 MATLAB 代码可生成所需滤波器：

```
fs = 11025;          %Sample frequency
fpass = 2000;        %Passband edge
fstop = 2700;        %Stopband edge
%Ripple must be specified in decibels
Rpass = 0.01;RpassDB = -20*log10(1-Rpass);
Rstop = 0.05;RstopDB = -20*log10(Rstop);
[N fn] = buttord(fpass/(fs/2), fstop/(fs/2), RpassDB, RstopDB);
[num den] = butter(N, fn);
[H f] = freqz(num, den, 1024, fs);
plot(f, abs(H));
```

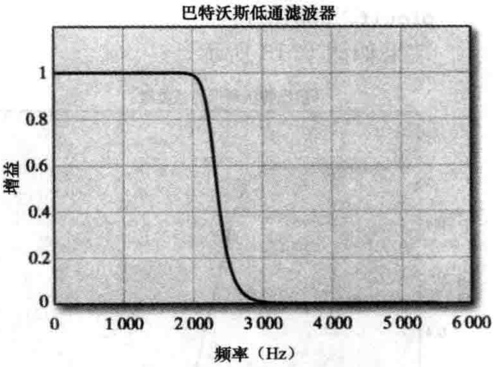


图 6.15 巴特沃斯低通滤波器，频带边界及波动满足要求

结果如图 6.15。

6.4.2 切比雪夫 I 型滤波器

切比雪夫 I 型低通滤波器也是全极点滤波器(all-pole filter)，且在许多方面与巴特沃斯滤波器类似。二者的主要区别在于通带及过渡带的形式。巴特沃斯滤波器具有最大平坦的通带及相对较宽的过渡带，如图 6.15 所示。而切比雪夫 I 型滤波器在通带具有等波动响应，同时改善了过渡带的宽度。在大部分情况下，若不需要特别平坦的通带，选择切比雪夫 I 型滤波器会使阶数更低，从而提高计算效率。

切比雪夫 I 型滤波器的幅度平方响应(amplitude-squared response)如下

$$H(s)H(-s) = \frac{1}{1+\varepsilon^2 V_N^2(f/f_{pass})} \tag{6.18}$$

其中，ε 是与通带波动相关的参数，f_{pass} 是通带边界频率，$V_N(x)$ 是切比雪夫多项式，其定

义如下

$$V_N(x) = \cos(N\arccos(x))$$

求解切比雪夫多项式，可以使用以下迭代关系[1]：

$$V_0(x) = 1$$
$$V_1(x) = x$$
$$V_{k+1}(x) = 2xV_k(x) - V_{k-1}(x)$$

由上几式，可罗列出切比雪夫多项式如下：

$$V_0(x) = 1$$
$$V_1(x) = x$$
$$V_2(x) = 2x^2 - 1$$
$$V_3(x) = 4x^3 - 3x$$
$$V_4(x) = 8x^4 - 8x^2 + 1$$
$$V_5(x) = 16x^5 - 20x^3 + 5x\cdots$$

波动参数 ε 与通带中的波动 R_p 相关，关系式如下

$$R_p = 1 - \frac{1}{\sqrt{1+\varepsilon^2}}$$

或

$$\varepsilon = \sqrt{\frac{1}{(1-R_p)^2} - 1}$$

因此可以看出，切比雪夫 I 型滤波器的极点落在 s 平面的椭圆上[1][2]。这个椭圆由如图 6.16 所示的两个圆来确定。椭圆的长短轴由以下公式得到

$$r_{\text{major}} = 2\pi f_{\text{pass}} \frac{C^2+1}{2C} \qquad (6.19)$$

以及

$$r_{\text{minor}} = 2\pi f_{\text{pass}} \frac{C^2-1}{2C} \qquad (6.20)$$

其中，

$$C = \left[\frac{1}{\varepsilon} + \sqrt{1+1/\varepsilon^2}\right]^{1/N} \qquad (6.21)$$

极点的角度与巴特沃斯滤波器极点的角度相同：$\pi(1+N+2k)/2N, k=0,1,\cdots,(2N-1)$。图 6.16给出了四阶切比雪夫 I 型滤波器的极点位置与角度。

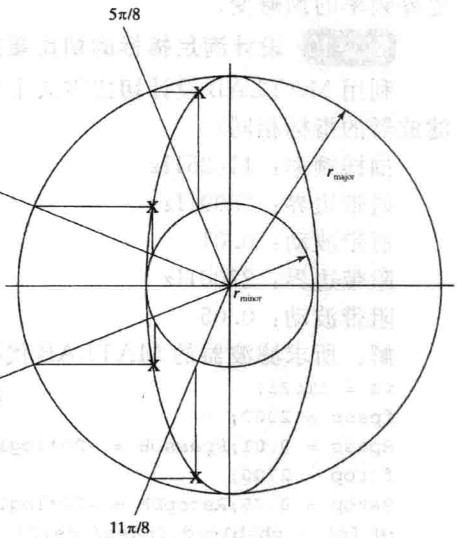


图 6.16　切比雪夫 I 型滤波器在 s 平面的极点分布，其中椭圆由半径为 r_{major} 及 r_{minor} 的圆确定

由图 6.16，可写出极点位置：

$$\theta_k = \pi(1+N+2k)/2N, k = 0,,1,\cdots,(2N-1)$$
$$x_k = r_{\text{minor}}\cos(\theta_k)$$
$$y_k = r_{\text{major}}\sin(\theta_k)$$

为了得到满足切比雪夫 I 型滤波器的最小阶数，需要求解式(6.18)中的 N。需注意，与巴特沃斯滤波器不同，切比雪夫 I 型滤波器的幅度平方函数(amplitude-squared equation)由通带边界频率来表示，而不是截止频率。因此，只需在阻带边界频率处(即 f_{stop})求解式(6.18)即可得到 N 值。f_{stop} 处的幅频响应为 R_{stop}。式(6.18)变为

$$R_{\text{stop}}^2 = \frac{1}{1+\varepsilon^2 V_N^2(f_{\text{stop}}/f_{\text{pass}})}$$

对式中 V_N^2 求解得

$$V_N^2(f_{\text{stop}}/f_{\text{pass}}) = \frac{1-R_{\text{stop}}^2}{R_{\text{stop}}^2\varepsilon^2} \tag{6.22}$$

切比雪夫多项式定义为

$$V_N(x) = \cos(N\arccos(x))$$

在这里，x 的值 $f_{\text{stop}}/f_{\text{pass}}$ 大于 1，因此切比雪夫多项式也可写为

$$V_N(x) = \cosh(N\text{arccosh}(x))$$

其中 $\cosh(x)$ 是双曲余弦函数，定义为

$$\cosh(x) = \frac{e^x + e^{-x}}{2} = \cos(jx)$$

替换式(6.22)求得 N 为

$$N = \frac{\text{arccosh}\left(\dfrac{(1-R_{\text{stop}}^2)^{\frac{1}{2}}}{R_{\text{stop}}\varepsilon}\right)}{\text{arccosh}(f_{\text{stop}}/f_{\text{pass}})} \tag{6.23}$$

需注意，若利用双线性变换将模拟滤波器转换为对应的数字滤波器，频率 f_{stop} 和 f_{pass} 应为预畸变后的频率。

式(6.23)可得到最小的 N 值。多数情况下这个值不是整数，需要向上取整。

MATLAB 中有两个函数，其函数名为 cheby1 与 cheb1ord，可以自动完成上述代数运算。若需要求解对应的数字滤波器，MATLAB 应用双线性变换前，需要进行通带和阻带边界频率的预畸变。

例 6-10　设计满足指标的切比雪夫Ⅰ型数字低通滤波器

利用 MATLAB 设计切比雪夫Ⅰ型数字低通滤波器，指标如下：（与例 6-9 中巴特沃斯滤波器的指标相同）

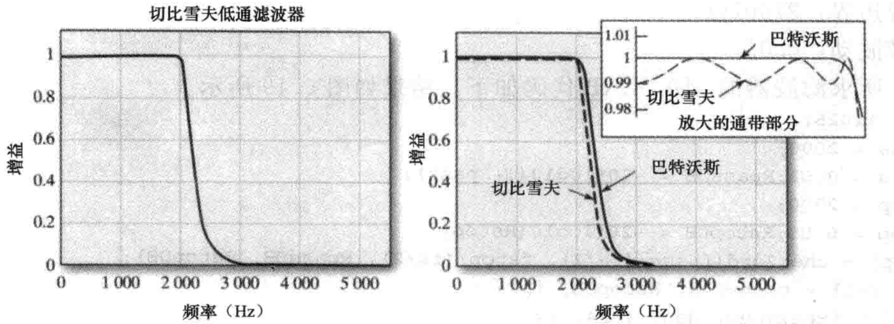
抽样频率：11025Hz
通带边界：2000Hz
通带波动：0.01
阻带边界：2700Hz
阻带波动：0.05

解：所求滤波器的 MATLAB 代码如下。

```
fs = 11025;
fpass = 2000;
Rpass = 0.01;RpassDB = -20*log10(1-Rpass);
fstop = 2700;
Rstop = 0.05;RstopDB = -20*log10(Rstop);
[N fp] = cheb1ord(fpass/(fs/2), fstop/(fs/2), RpassDB, RstopDB);
[num den] = cheby1(N, RpassDB, fp);
[H f] = freqz(num, den, 1024, fs);
plot(f, abs(H));
```

结果如图 6.17 所示。

通过观察上面两例，可以比较巴特沃斯滤波器及切比雪夫Ⅰ型滤波器的设计过程。设计巴特沃斯滤波器时，截止频率是一个重要参数。对于低通滤波器，截止频率通常为增益是 $\sqrt{2}/2=0.707$ 处的频率值，这适用于巴特沃斯滤波器。而对于切比雪夫滤波器，重要的参数则是通带的边界频率(f_{pass})。由于在通带确定了波动 R_{pass}，且在 0Hz 处为单位增益 1，所以在频率 f_{pass} 处增益为 $1-R_{\text{pass}}$。因此，若已知通带和阻带的边界频率和波动，则设计



a）切比雪夫低通滤波器，注意通带波动

b）满足相同指标的切比雪夫滤波器，其与巴特沃斯滤波器相比较，切比雪夫滤波器的过渡带更窄

图 6.17　例 6-10 图

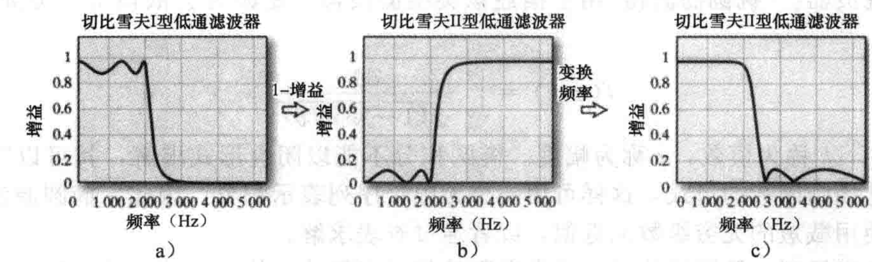
巴特沃斯滤波器时，需要先确定其截止频率。

在对比巴特沃斯滤波器与切比雪夫Ⅰ型滤波器时，需要提及的另一个差别是最终结果。切比雪夫Ⅰ型滤波器在通带有较明显的波动，但其满足设计指标的最低阶数为 6，而巴特沃斯却是 10。在大多应用中，若不需要绝对平坦的通带，可选择切比雪夫滤波器以得到更低的阶数。

6.4.3　切比雪夫Ⅱ型滤波器(逆切比雪夫滤波器)

切比雪夫Ⅰ型滤波器也可以相反的形式构建。在这种情况下，通带和阻带的特性将发生改变。也就是说，通带最大化平坦(或单调)而阻带有波动。这种相反形式的切比雪夫滤波器常称为切比雪夫Ⅱ型滤波器。

切比雪夫Ⅱ型的构建遵循Ⅰ型。图 6.18a 给出了一个 5 阶的切比雪夫Ⅰ型滤波器的幅频响应。可看到，通带有明显的波动，而阻带是单调变化。用 1 减去这一幅频响应，新的响应如图 6.18b 所示。其中通带单调变化而阻带有波动，但已由原来的低通滤波器变为高通。若要得到切比雪夫Ⅱ型低通滤波器，需对频率作重映射，使低频映射到高频，反之亦然。此过程可以将原方程的 f 由 $1/f$ 代换来完成。如图 6.18c 即为希望得到的幅频响应。



a）　b）　c）

图 6.18　切比雪夫Ⅱ型低通滤波器的构建

例 6-11　设计切比雪夫Ⅱ型低通滤波器

利用 MATLAB 设计切比雪夫Ⅱ型数字低通滤波器，需满足以下指标：(与例 6-9 巴特沃斯滤波器的要求相同。)

抽样频率：11025Hz

通带边界：2000Hz

通带波动：0.01

阻带边界：2700Hz

阻带波动：0.05

解：所求滤波器的MATLAB代码如下。结果如图6.19所示。

```
fs = 11025;
fpass = 2000;
Rpass = 0.01;RpassDB = -20*log10(1-Rpass);
fstop = 2700;
Rstop = 0.05;RstopDB = -20*log10(Rstop);
[N fp] = cheb2ord(fpass/(fs/2), fstop/(fs/2), RpassDB, RstopDB);
[num den] = cheby2(N, RstopDB, fp);
[H f] = freqz(num, den, 1024, fs);
plot(f, abs(H));
```

6.4.4 椭圆滤波器

巴特沃斯和切比雪夫模拟低通滤波器都是“全极点”滤波器(切比雪夫Ⅱ型不是)。当利用双线性变换将N阶的模拟滤波器转换到对应的数字滤波器后，该数字滤波器的传输函数在$z=-1$处(对应模拟滤波器的$f_s/2$处)会有N个零点，这些零点的位置导致了一个单调变化的阻带。而在通带中，巴特沃斯滤波器的极点分布方式使通带达到最大平坦。对于切比雪夫Ⅰ型滤波器，通带会有一定的波动，极点的分布使其过渡带比巴特沃斯滤波器的更窄。椭圆模拟低通滤波器在通带和阻带都存在波动，使得过渡带比前两个滤波器都窄。这是因为椭圆滤波器既有极点也有零点，将二者放置很近时，可使得滤波器的过渡带更陡峭。

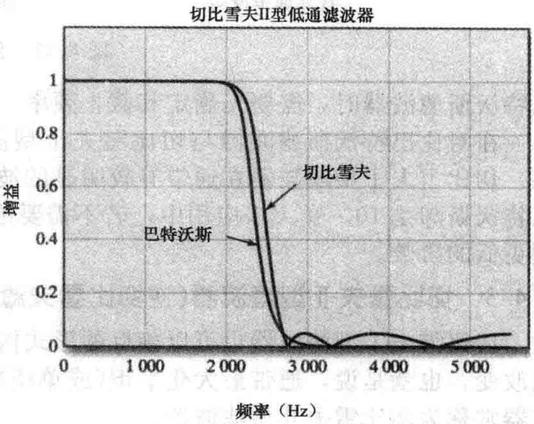


图6.19 6阶切比雪夫Ⅱ型滤波器，其与满足同样指标的巴特沃斯滤波器相对比

椭圆滤波器是Wilhelm Cauer为德国电话行业所研发的。因此，椭圆滤波器有时也称为Cauer滤波器。“椭圆滤波器”用于描述该类型滤波器，是因为它依赖第一类椭圆积分，形如[3]

$$f(\varphi,k)=\int_0^{\varphi}\frac{\mathrm{d}\theta}{\sqrt{(1-k\sin^2\theta)}}$$

上式中，k称为模数，φ称为幅度。椭圆积分不能以闭合形式求解，其可以写成一种称为雅可比椭圆函数的形式，这样可用一个无限长序列表示积分。因此，椭圆滤波器的设计依赖于使用截短的无穷级数来近似，或者通过查表求解。

前面已经提到，椭圆滤波器在通带和阻带都具有等波动特性。等波动是指频带中波动的幅度是均匀的。在FIR滤波器中已经知道，这种等波动特性是某种意义上优化后的滤波器属性。在满足设计指标的前提下，椭圆滤波器需要的阶数是最低的。

椭圆滤波器的难点在于设计。其研发与设计在数学上很复杂，完整的流程已经超出本书的范围。本书借鉴了由另外一些人[2][4]提出的经过严格推导的直观方法。

巴特沃斯或切比雪夫低通滤波器的幅度平方函数的一般形式为：

$$A^2(\omega)=\frac{1}{1+\varepsilon T^2(\omega)}$$

式中，ε是常量，$T(\omega)$是有理函数。

该式中，当 $T(\omega)\to 0$ 时，幅度函数趋近于1。同样，$T(\omega)\to\infty$ 时，幅度函数趋近于0。若将 $T(\omega)$ 写为两个多项式之比的形式，则显然 $T(\omega)$ 的极点为幅度函数的零点。

$$A^2(\omega)=\frac{1}{1+CN^2(\omega)/D^2(\omega)}=\frac{D^2(\omega)}{D^2(\omega)+CN^2(\omega)} \tag{6.24}$$

图6.20给出了在通带和阻带都具有等波动的低通滤波器。由于这是椭圆滤波器的特征，因此可以将此图与幅度平方函数(6.24)相关联，从而推导出椭圆滤波器的传输函数。

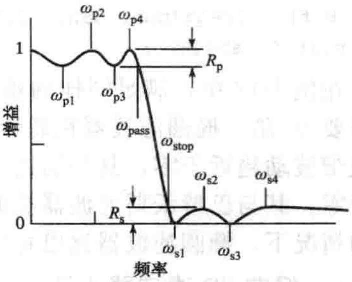


图6.20 具有等波动特性的低通滤波器

由式(6.24)可知，$D(\omega)$ 的根一定在 ω_{s1} 及 ω_{s2} 等处，而 $N(\omega)$ 的根一定在0、ω_{p2}、ω_{p4} 等处。同时也可知函数在 ω_{p1} 和 ω_{p3} 点的值为 $(1-R_p)$，在 ω_{s2} 和 ω_{s4} 点的值为 (R_s)。若对式(6.24)求导，从图6.20中可看出，在函数斜率为0的点 $(0, \omega_{p1}, \omega_{p2}, \omega_{p3}, \omega_{p4}, \omega_{s2}$ 及 $\omega_{s4})$ 其导数为0。组合这些信息，即可写出滤波器传输函数对应的微分方程。该微分方程以积分形式给出时即为第一类椭圆积分。这类积分没有闭合形式的解，因此需要查表或利用无穷级数近似。这些关于椭圆滤波器的推导和近似超出了本书的范围[2]。

MATLAB利用两个函数实现椭圆滤波器的设计：ellipord用来确定满足指标的适当阶数，ellip用来构建椭圆滤波器。

例6-12 设计椭圆数字低通滤波器

利用MATLAB设计椭圆数字低通滤波器，需满足以下指标：(与例6-9巴特沃斯滤波器的指标相同。)

抽样频率：11025Hz

通带边界：2000Hz

通带波动：0.01

阻带边界：2700Hz

阻带波动：0.05

解： 所求滤波器的MATLAB代码如下。结果如图6.21所示。

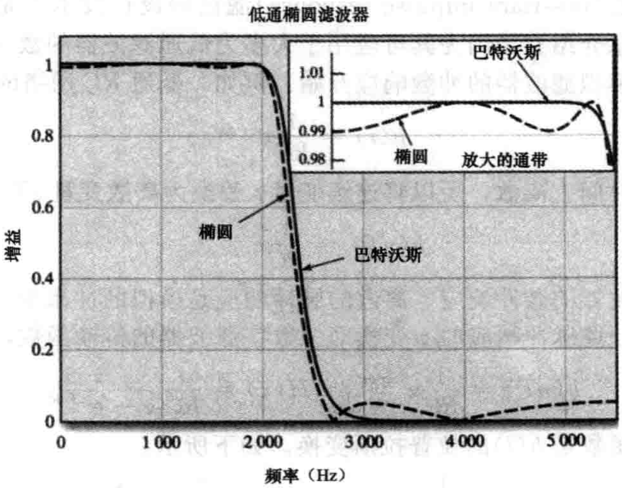


图6.21 低通椭圆滤波器，其与满足同样指标的巴特沃斯滤波器相对比

```
fs = 11025;
fpass = 2000;
Rpass = 0.01;RpassDB = -20*log10(1-Rpass);
fstop = 2700;
Rstop = 0.05;RstopDB = -20*log10(Rstop);
[N fp] = ellipord(fpass/(fs/2), fstop/(fs/2), RpassDB, RstopDB);
[num den] = ellip(N, RpassDB, RstopDB, fp);
[H f] = freqz(num, den, 1024, fs);
plot(f, abs(H));
```

在例 6-12 中，满足同样的指标时，椭圆滤波器所需阶数为 4 阶，而巴特沃斯滤波器却需要 10 阶。椭圆滤波器在通带和阻带都不单调，且有非常窄的过渡带。若椭圆滤波器的阻带波动趋近于零，其与切比雪夫滤波器类似。若椭圆滤波器的通带和阻带的波动都趋近于零，其与巴特沃斯滤波器类似。一般地，若允许通带和阻带存在波动，在满足相同指标的情况下，椭圆滤波器比巴特沃斯和切比雪夫滤波器的计算效率都要高。

6.4.5　经典 IIR 滤波器小结

尽管有许多类型的 IIR 数字滤波器，但本书介绍的 4 种经典滤波器(巴特沃斯、切比雪夫Ⅰ型、切比雪夫Ⅱ型和椭圆)是最普遍的。表 6.4 总结了这些滤波器的特点。

表 6.4　经典滤波器的特点

滤波器类型	通带	过渡带	阻带	说明
巴特沃斯	最大平坦	宽	单调	低通在 $fs/2$ 处有 N 个零点，设计过程简单，满足给定指标需要相对高的阶数 阻带衰减比所需的衰减更多
切比雪夫Ⅰ型	等波动	中	单调	低通在 $fs/2$ 处有 N 个零点，设计过程简单，满足给定指标需要的阶数比椭圆高，比巴特沃斯低 阻带衰减比所需的衰减更多
切比雪夫Ⅱ型	单调	中	等波动	设计过程比Ⅰ型稍复杂，阶数与Ⅰ型相同
椭圆	等波动	窄	等波动	设计过程比较复杂。相同指标下，比巴特沃斯和切比雪夫的阶数低

6.4.6　脉冲响应不变

脉冲响应不变法(invariant impulse response)滤波器设计技术不是“经典的”数字滤波器的设计方法。本章介绍它是因为其可应用于大多为低通滤波器的数字控制系统。

该设计方法从模拟滤波器的冲激响应开始。例如，低通 RC 网络的冲激响应为

$$h(t) = \frac{1}{RC}\mathrm{e}^{-t/RC}$$

因为数字滤波器在时间上离散，所以将连续变量 t 替换为离散变量 nT。则上式改为

$$h(nT) = \frac{1}{RC}\mathrm{e}^{-nT/RC}$$

这是对应的数字滤波器的脉冲响应。离散的脉冲响应是模拟的冲激响应的近似，其只在离散时间点上有定义。此脉冲响应的 z 变换即为数字滤波器的传输函数，即

$$h(nT) = \frac{1}{RC}\mathrm{e}^{-nTRC} \Leftrightarrow / H(z) = \frac{1}{RC}\frac{z}{z-\mathrm{e}^{-T/RC}}$$

模拟滤波器的传输函数是 $h(t)$ 的拉普拉斯变换。如下所示。

$$h(t) = \frac{1}{RC}\mathrm{e}^{-tRC} \Leftrightarrow / H(s) = \frac{1}{RC}\frac{s}{s+1/RC} \tag{6.25}$$

若取 $R=1\text{k}\Omega$ 和 $C=1/\mu\text{F}$，则可利用 MATLAB 绘出模拟滤波器的频率响应。而对于

数字滤波器，需要选择一个合适的抽样周期 T。根据抽样定理，频谱的有效部分为 0Hz～$f_s/2$，这里抽样频率取 $f_s=1000\text{Hz}(T=0.001\text{s})$。根据该 T 值可绘出数字滤波器的频率响应如图 6.22 所示。比较数字滤波器与模拟滤波器的频率响应，可以看到在低频段几乎没有失真。而当频率接近 $f_s/2$ 时，数字滤波器的增益明显比模拟滤波器要大。

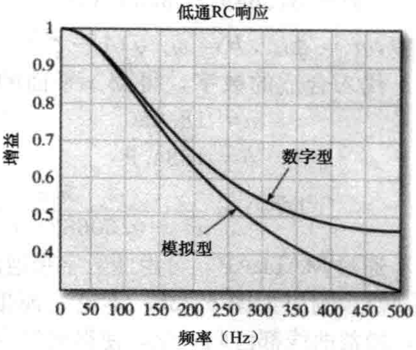


图 6.22　数字和模拟的低通 RC 网络的频率响应，应用脉冲响应不变法

前面对 RC 电路的分析让人不禁想问，脉冲响应不变法究竟有多好？可以用以下分析来解答：考虑任意连续函数 $g(t)$，以及对其抽样后的函数 $g_s(nT)$。若 $g(t)$ 频谱为 $G(\omega)$，为了方便观察抽样过程如何改变了频谱，我们将尝试利用 $G(\omega)$ 来表示抽样后信号的频谱。为达此目标，先用 $g(t)$ 表示 $g_s(nT)$ 并对结果进行傅里叶变换。

$$g_s(nT)=\sum_{n=-\infty}^{\infty}g(t)\delta(t-nT) \tag{6.26}$$

时域的乘法对应频域的卷积，对上式进行傅里叶变换可得

$$F\{g_s(nT)\}=\frac{1}{2\pi}G(\omega)*F\left\{\sum_{n=-\infty}^{\infty}\delta(t-nT)\right\} \tag{6.27}$$

对冲激函数(delta function)的无穷级数进行傅里叶变换[5]可得

$$F\left\{\sum_{n=-\infty}^{\infty}\delta(t-nT)\right\}=\omega_s\sum_{k=-\infty}^{\infty}\delta(\omega-k\omega_s),\quad \omega_s=\frac{2\pi}{T} \tag{6.28}$$

将式(6.28)代入(6.27)，并由连续卷积(continuous convolution)定义得

$$F\{g_s(nT)\}=\frac{1}{T}\int_{-\infty}^{\infty}G(\lambda)\sum_{k=-\infty}^{\infty}\delta(\omega-\lambda-k\omega_s)\mathrm{d}\lambda \tag{6.29}$$

式(6.29)求和变量是 k，积分变量是 λ，因此改变积分和求和的顺序方程仍保持不变。

$$F\{g_s(nT)\}=\frac{1}{T}\sum_{k=-\infty}^{\infty}\int_{-\infty}^{\infty}G(\lambda)\cdot\delta(\omega-\lambda-k\omega_s)\mathrm{d}\lambda \tag{6.30}$$

又因 $\int_{-\infty}^{\infty}f(x)\delta(x-k)\cdot\mathrm{d}x=f(k)$，式(6.30)可简化为

$$F\{g_s(nT)\}=\frac{1}{T}\sum_{k=-\infty}^{\infty}G(\omega-k\omega_s) \tag{6.31}$$

式(6.31)即为信号 $g(t)$ 的频谱 $G(\omega)$ 表示的抽样后信号 $g_s(nT)$ 的频谱，此结果与抽样定理推导出的结果吻合。因此，若 $G(\omega)$ 是带限的，即 $\omega\geqslant\omega_s/2$ 时 $G(\omega)=0$，抽样后系统的频谱与原系统频谱相同，主要差别有两个，一是两者幅度相差一个常数，抽样后系统的频谱以 ω_s 为周期不断重复。对于非带限的滤波器，如高通和带阻滤波器，此方法构建的滤波器会有严重的混叠现象。

在利用脉冲响应不变法时，式(6.31)对于抽样频率的选择给出了依据。即抽样频率应使得在 $\omega>\omega_s/2$ 时，模拟系统的频率响应幅度趋近于零。

例 6-13　利用脉冲响应不变法转换带通滤波器

某带通滤波器的传输函数如下，利用脉冲响应不变法求出近似等价的数字滤波器。

$$H(s)=\frac{k}{s^2+37.649s+3.5531\times10^6}$$

解： 利用 MATLAB 绘出 $H(s)$ 对应的幅频响应，如图 6.23a 所示。选择适当的增益常数 k 使得通带中间的增益为 1。由幅频响应可知，带通滤波器的通带中心在 300Hz 处。

若认为该模拟带通滤波器的增益在500Hz处趋近于零，则较合适的抽样频率应为1000Hz。

查阅第3章拉普拉斯变换表和 z 变换表，得到与 $H(s)$ 相应的 $H(z)$。

$$\frac{k}{s^2+37.649s+3.5531\times10^6}=\frac{k}{s^2+2\zeta\omega_n s+\omega_n^2}\Rightarrow\frac{(1/b)[ze^{-aT}\sin(bT)]}{z^2-2ze^{-aT}\cos(bT)+e^{-2aT}}$$

其中，$a=\zeta\omega_n$，$b=\omega_n\sqrt{(1-\zeta^2)}$

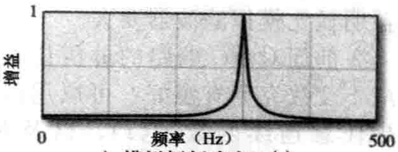
代入合适的数字，得到 z 平面的传输函数。

$$a=18.850$$

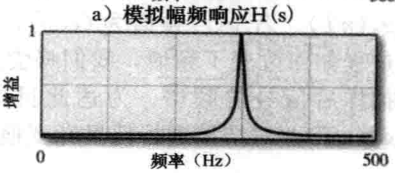
$$b=1884.9$$

$$H(z)=\frac{k_1 z}{z^2+0.60632z+0.96300}$$

通过MATLAB，可由此数字带通滤波器的传输函数得到其幅频响应如图6.23b。不论是模拟还是数字的幅频响应，增益曲线都已归一化，使得通带中间的增益为1。



a）模拟幅频响应H(s)



b）数字幅频响应H(z)

图6.23 模拟和数字幅频响应

MATLAB的函数impinvar可以实现脉冲响应不变法。若给定 s 域的传输函数和抽样频率，该函数便可得到对应的 z 域传输函数。下面给出的MATLAB代码，可以实现前面的计算结果。

```
nums = 71000;
dens = [1 37.649 3.5531e06];
[Hs Ws] = freqs(nums,dens,512);
fa = Ws/(2*pi);
figure(1);clf;
subplot(2,1,1)
plot(fa,abs(Hs))
axis([0 500 0 1])
fs = 1000;
[numz denz] = impinvar(nums,dens,fs);
[H f] = freqz(numz,denz,512,fs);
subplot(2,1,2)
plot(f,abs(H))
axis([0 500 0 1])
```

◀

从例6-13可以看到，稳定的模拟滤波器可转换为稳定的数字滤波器。很容易证明这具有一般性。模拟滤波器的传输函数可用部分分式法展开

$$H(s)=\sum_{i=1}^{N}\frac{K_i}{s+P_i}$$

其中，P_i 可以为复数。若对此展开式使用脉冲响应不变法，得到

$$\frac{K}{s+P}\Rightarrow\frac{Kz}{z-e^{-PT}}$$

其中，T 是抽样周期。若 P 是复数且 $H(s)$ 是稳定滤波器，则有

$$s=-P=-\sigma-j\omega$$

为了满足稳定性，$\sigma>0$ 以使极点都位于 s 平面的左半平面。这意味着 $|e^{-(\sigma+j\omega)T}|<1$，即数字滤波器的极点都位于 z 平面单位圆内。因此，对于脉冲响应不变法，s 域稳定的滤波器必然导致 z 域滤波器也是稳定的。脉冲响应不变法应用到巴特沃斯、切比雪夫和椭圆低通滤波器时，可以得到很好的结果。

6.5 IIR滤波器在 z 平面的极点和零点

通过 $e^{j\omega T}\to z$ 代入系统的传输函数，可以计算并绘制数字滤波器的幅频响应和相位响应。对于较高阶的滤波器，计算将会变得较为复杂，即使借助计算机也需要相当的时间。一个更

快的方法是，通过增益和相位曲线，检测滤波器的极点和零点的分布。这里描述的对增益和相位曲线近似的方法基于传输函数中向量的几何解释。图 6.24 显示了单位圆的一部分，其中有一个极点 P_1。在传输函数中，这个极点表现为项 $(z-P_1)$。从几何上看，可以从原点到极点做一条向量，其长度为 $|P_1|$。若要计算传输函数在单位圆上一点 z_i 的值，可以从原点到该点做第二条向量。这条向量的长度为 $|z_i|=1$ 且角度为 θ_i。从 P_1 到 z_i 再做一条向量，如图 6.24 所示，这条向量的长度为 $|z_i-P_1|$，角度为 θ。

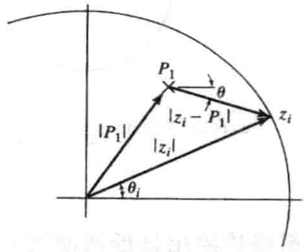


图 6.24 传输函数中各项的几何说明

因此从几何上，通过从每个极点和零点到所求点做向量的方法，可以计算出单位圆上的任何一个取值 z 的传输函数值。所有零点向量长度的乘积除以所有极点向量长度的乘积就得到传输函数在某一点的幅度。z 平面上角度 θ_i 实际就是数字域角频率 Ω，由数字域角频率 θ_i 可求出对应的模拟域频率 f：$\theta_i=\omega T=2\pi f/f_S$。同样地，可以求出分子向量角度之和与分母向量角度之和的差，即为传输函数在取值点的相位角。

通过从几何角度观察相位角的计算，可以很清楚地知道为什么在单位圆上的零点会引起相位曲线上的不连续：动点 z_i 从一边向单位圆上的零点移动时产生一个角度 θ，在通过零点时，这个角度会产生 180°的跃变，从而导致不连续。

尽管可以精确地计算不同向量的长度以及角度以求出频率响应，但这种方法在计算上还是很繁琐，并不比其他方法更优。它的优点在于通过设想 z_i 在单位圆上运动时不同向量的变化，可以使近似结果形象化。以下是“经验法则”。

近似频率响应的规则。

幅度曲线：

1. 确定传输函数中 $z=1(e^{j0}=1)$的点作为起始点。
2. 确定传输函数中 $z=-1(e^{j\pi}=(-1))$的点作为终点。
3. 确定零点的频率。对于所有位于单位圆上的零点，标记其对应的幅值等于 0。
4. 当 z_i 在单位圆上从 $z_i=1$ 移动到 $z_i=-1$ 变化时，分析极点向量和零点向量长度的变化。当分母向量的长度减小时(z_i 接近极点)，幅度会增大。当分子向量趋近 0 时，幅度会减小。

相位曲线：

1. 确定传输函数中 $z=1$ 的点作为起始点。若在 $z=+1$ 处没有零点，起始相位角在幅度为正时为 0°，在幅度为负时为 180°。若在 $z=+1$ 处有 1 个或多个零点，起始相位角为$\pm N\cdot 90°$，其中，N 是 $z=+1$ 处零点的个数，符号由 $z=+1$ 附近幅度的符号决定
2. 确定传输函数中 $z=-1$ 的点作为终点。最终的角度值可参考步骤 1 来计算。
3. 确定零点的频率。当分子向量靠近零点时，会使传输函数增加一个角 α。当点 z_i 经过零点后，角 α 将变化 180°。这导致了相位曲线的不连续。
4. 当 z_i 在单位圆上从 $z_i=1$ 移动到 $z_i=-1$ 时，分析极点向量和零点向量的角度变化。记得要用分子向量角之和减去分母向量角之和。

例 6-14 估计幅频响应

利用前面所述的近似法求解图 6.25 所示的近似幅度曲线。

解：(a) 若令 $z=1$，可得到在频率 0Hz 处增益为 1。若另 $z=-1$，可得到在频率 $f_S/2$ 处增益为 0。设想一个沿着单位圆移动的取值点(evaluation point)，以及由极点和零点指向该点的向量。当取值点靠近极点时，该极点的分母向量变短而分子向量变化很小。这导致增益在该极点频率处增大。极点大约在 60°处，相应的频率为 $f_S/6$。取值点经过极点后，分母向量开始变长而分子向量开始变短，直到 $z=-1$ 处，分子为 0。由以上信息，可



a) $H(z)=0.36\dfrac{z+1}{(z^2-0.8z+0.52)}$ b) $H(z)=0.1546\dfrac{z^5-3.66z^4+6.39z^3-6.39z^2+3.66z-1}{z^5-0.661z^4+1.24z^3-0.272z^2+0.276z-0.0206}$

图 6.25 IIR 滤波器的极点/零点用以估算频率响应

粗略描绘出该低通滤波器的幅频响应。结果如图 6.26 所示。

(b) 因为在 $z=+1$ 处有一个零点，所以在频率 0Hz 处增益为 0。若将 $z=-1$ 代入传输函数，可得在 $f_s/2$ 处增益为+1。在约为 45°与 60°处还有两个零点，对应频率为 $f_s/8$ 与 $f_s/6$。当取值点通过第二个零点后，增益会急剧上升，因为分子向量不断增大而分母向量由于附近的极点而减小。越过极点后(约在 $3f_s/8$ 处)，增益会保持不变(有一些波动)直到移动到点 $z=-1$处。图 6.27 给出了该高通滤波器的粗略幅频响应图，在 $f_s/2$ 处增益为 1。

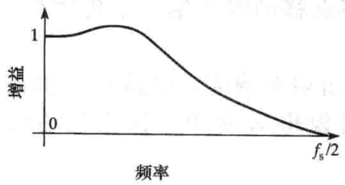


图 6.26 (a)部分低通滤波器的粗略图

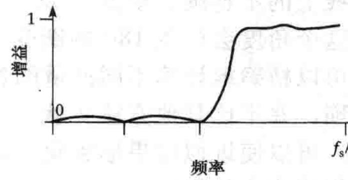


图 6.27 (b)部分高通滤波器粗略图

IIR 滤波器的零极点位置小结

图 6.24 利用几何解释说明了传输函数的频率响应，由此可以得出 IIR 滤波器在 z 平面关于极点和零点位置的如下结论。

- 为使得系统稳定，所有的极点都应位于 z 平面单位圆内。零点不影响稳定性，可以在 z 平面的任何位置。
- 为使得系统因果，极点的数量不少于零点的数量。
- 若使差分方程有实系数，极点和零点在 z 平面要关于实轴对称。每个复数项都有其对应的共轭项。
- 通过在单位圆上取不同 z 值求得频率响应。单位圆上的零点会使该点的幅度响为零。
- 单位圆上的零点会使相位曲线不连续，因为取值点经过该点时零点向量的方向会改变 180°。
- 除了在原点的极点，其他极点会导致差分方程含反馈项。任何单位圆上的极点都会导致单位增益的反馈项，并产生振荡。
- 因为极点和零点均关于实轴对称，所以复数项在相位曲线 0Hz 处没有影响，因为任何向量的相角都与其共轭项的相角相抵消。因此，若在 $z=+1$ 处没有零点，相位曲线的初始相位必为 0°或 180°。若在 $z=+1$ 处有零点，则会使相位曲线加 90°。所有 IIR 滤波器的相位曲线的初始相位必为 90°的整数倍。在 $f_s/2$ 处也有相同情况。
- 在幅度曲线中，极点距离单位圆越近，在相应频率处增益上升越大。

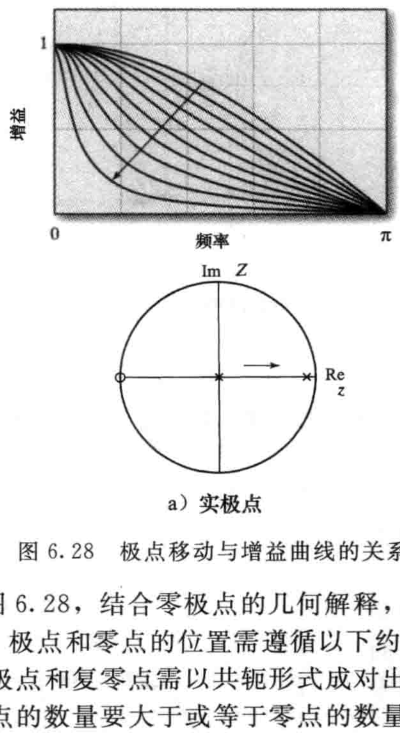
6.6 IIR 滤波器的直接设计

在前面章节中，IIR 滤波器的设计都是由已知的模拟滤波器通过双线性变换法转换到 z

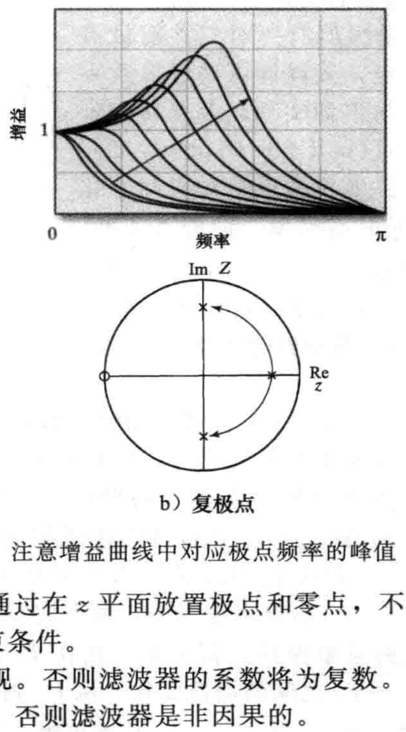
平面。不通过模拟滤波器，同样也可以设计 IIR 数字滤波器，这种设计方法称为"直接"设计。若有合适的软件提供即时反馈，设计者可通过在 z 平面放置极点与零点的方式，可以简单地设计出低阶的 IIR 数字滤波器。同样地，通过计算器就可设计二阶 IIR 滤波器，如陷波滤波器(notch filter)与谐振器(resonator)。最后，更高阶的数字滤波器可通过计算机求解联立方程而得到。本书主要介绍 Pade 方法、Prony 方法，以及利用 Yule-Walker 方程设计的方法。

6.6.1 放置零极点的设计方法

在 6.5 节中，通过极/零点图的几何解释来求解传输函数的频率响应。即，在求解滤波器的幅度和相位曲线时，设想一取值点在 z 平面沿单位圆移动，同时分析传输函数中的极点与零点到该点的向量。滤波器的增益为该传输函数所有零点向量的乘积除以所有极点向量的乘积。按照这个思路，观察极点到单位圆上点的向量在长度和相位的变化，就可以形象地理解极点对增益的影响。显然，在原点处的极点不影响幅度曲线，因为从原点到单位圆的向量长度总为 1。另一方面，与单位圆较近的极点，当取值点与之接近时，向量的长度会变小，由于极点向量的长度在分母部分，所以与单位圆相近的极点在其角度对应的频率处会产生较高的增益。图 6.28 利用一个简单的传输函数说明了此过程。其中给出了极点移动时相应增益曲线的变化，图 a 对应实极点，b 对应复极点。这两种情况的幅度曲线都经过了归一化，在频率为 0 处增益为 1。对于在实轴的极点，增益曲线在 0Hz 处最高。当极点离开实轴成为复数时，增益曲线的峰值出现在极点角度对应的频率处。



a）实极点



b）复极点

图 6.28 极点移动与增益曲线的关系，注意增益曲线中对应极点频率的峰值

根据图 6.28，结合零极点的几何解释，通过在 z 平面放置极点和零点，不难构建不同的滤波器。极点和零点的位置需遵循以下约束条件。

1. 复极点和复零点需以共轭形式成对出现。否则滤波器的系数将为复数。
2. 极点的数量要大于或等于零点的数量。否则滤波器是非因果的。
3. 若使滤波器稳定，极点不能位于 z 平面单位圆外。
4. 为了满足第 2 条约束条件，可在原点放置极点。原点处的极点不影响幅度曲线。

这种方法只局限于低阶滤波器，因为高阶滤波器具有较多的零极点，零极点向量数量较多时交互会更加困难。该方法非常简单而且易于应用。为了使该方法更加有效，可通过计算机交互地显示极点或零点移动的增益曲线。习题 6.45 说明了如何编写 MATLAB 代码使此过程简单化。

6.6.2 二阶谐振器及陷波滤波器的设计

如图6.29所示的RLC电路，若选择合适的R、L和C值，使得系统传输函数出现复共轭极点，则称该电路是谐振的。描述该系统的传输函数如下

$$\frac{V_o(s)}{V_i(s)} = \frac{s(R/L)}{s^2 + s(R/L) + 1/(LC)} \tag{6.32}$$

式中，$R=100\Omega$，$L=10\text{H}$，$C=1\mu\text{F}$。谐振频率在$1/(LC)$处。增益经过归一化，峰值为1。

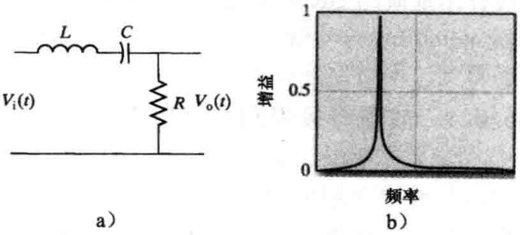


图6.29 RLC谐振电路及增益曲线

通过对式(6.32)进行双线性变换，可以得到对应的数字谐振器。其z域传输函数的形式为

$$H(z) = \frac{z^2 - 1}{z^2 + az + b} \tag{6.33}$$

上式中，选择a和b的值使传输函数出现复共轭极点，可得到如图6.30所示的典型的零极点分布图及其增益曲线。双线性变换法使得z平面的零点出现在$z=+1$和$z=-1$处，所以增益曲线在端点处都为0。其中，谐振的峰值对应极点的角度$\theta=\omega T$。

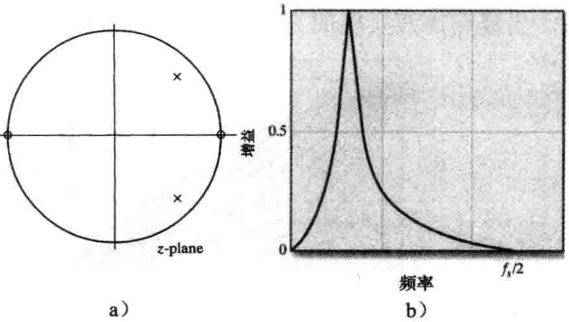


图6.30 数字谐振器的极/零点图及其对应的增益曲线

由于此谐振器有一对复共轭极点和两个实零点，其设计可通过放置零极点的方法简单地实现。谐振器的带宽由半功率点(half power point)之间的频带确定。半功率点是对应归一化增益为$\sqrt{2}/2$处的频率。回顾6.5节，借助极点和零点到单位圆上对应频率点的向量，可求出给定频率点的增益。图6.31对数字谐振器的增益曲线做出了几何解释。

该图中，单位圆上对应峰值的点与极点P_1的角度一致，标为f_P。若有极点位于$r\angle\theta$处，则该极点到单位圆上点f_P的距离为$1-r$。因此，该极点对于峰值点处增益曲线的贡献是$1-r$。图6.31中，点f_L是带宽中较低频率的半功率点，而f_U是带宽中较高频率的半功率点。若假设在$f_L \sim f_U$小频率范围内，两个实零点和复共轭极点的贡献保持近似常数，则在P_1处的极点主导$f_L \sim f_U$之间的增益变化。从P_1到频率f_L处的向量长度为$\sqrt{2}(1-r)$。从而可得

$$\tan(\theta_B/2) = 1 - r$$

对于具有很陡峰值的谐振器，θ_B为数字域角频率带宽，其数值将很小。而若x很小，存在$\tan(x) \approx x$。故

$$\theta_B \approx 2(1-r)2\pi f/f_s$$

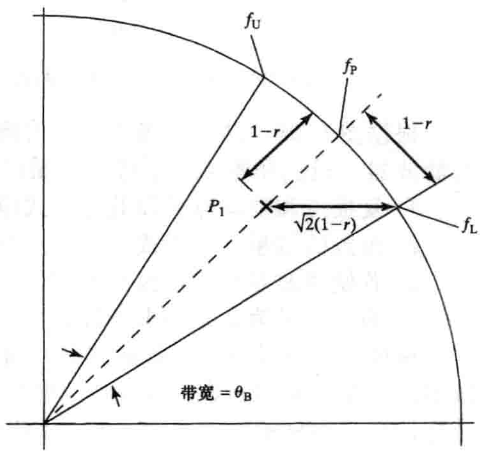


图6.31 几何说明数字谐振器的频率响应

根据数字域角频率与模拟频率之间的对应关

系 $\theta=\omega T=2\pi f/f_S$，可得到 θ_B 所对应的模拟域频带宽度 f_B，即

$$\text{带宽} = f_B \approx \frac{f_S(1-r)}{\pi}\text{Hz} \tag{6.34}$$

例 6-15 设计一个数字谐振器

设计一个抽样频率 $f_S=10000\text{Hz}$ 的数字谐振器，带宽为 125Hz，中心频率为 1000Hz。

解：将带宽和抽样频率代入式(6.34)得

$$125 = \frac{10000(1-r)}{\pi}$$

解上式得 $r=0.96073$，$r^2=0.92300$。

此极点的角对应于中心频率 1000Hz 处，极点应位于 $\frac{2\pi\times 1000}{10000}=0.62832\text{rad}=36°$。

在笛卡儿坐标系(Cartesian coordinates，即直角坐标系)中，极点在 $0.077725\pm j0.56470$ 处。该谐振器的传输函数为

$$H(z) = K\frac{z^2-1}{z^2-1.5545z+0.92300}$$

其中，K 用来归一化峰值点的增益。

以下 MATLAB 代码可绘出图 6.32 所示的增益曲线。

```
fs = 10000;
num = [1 0 -1];
den = [1 -1.5545 .92300];
[H f] = freqz(num,den,512,fs);
biggest = max(abs(H));
plot(f, abs(H)/biggest);
```

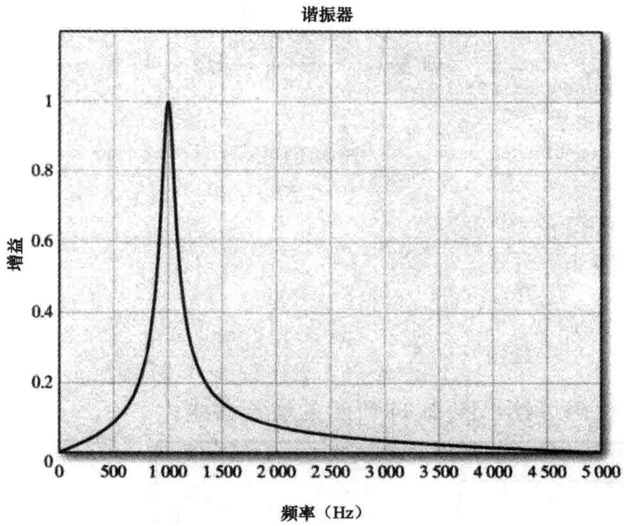
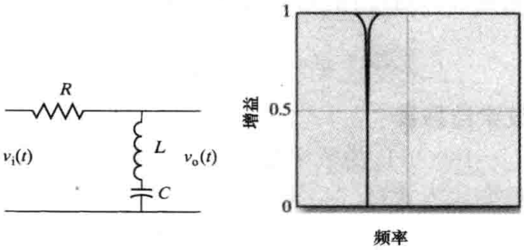


图 6.32 数字谐振器的增益曲线 ◀

重新排列图 6.29 中的 *RLC* 网络即可构建陷波滤波器，如图 6.33 所示。陷波滤波器可以去除输入信号中很窄的一个频带，其常用来去除已知频率的噪声，如来自电力线中的 60Hz 干扰。

RLC 电路的传输函数如下

$$H(s) = \frac{s^2+1/(LC)}{s^2+(R/L)s+1/(LC)}$$



a) b)

图 6.33 具有陷波滤波器特性的串联谐振电路

对上式利用双线性变换法可得到其对应的 z 域传输函数

$$H(z) = K\frac{z^2 + b_1 z + b_2}{z^2 + a_1 z + a_2}$$

其中，K 用来归一化最大增益。方程的系数与一组复零点及复极点相关。零点都在单位圆上，而极点都在单位圆内，零点与极点成对出现，且具有相同的角频率。极点与零点的距离控制滤波器的带宽。容易证明，若极点在 $r\angle\theta$ 处，其带宽可利用谐振器同样的方程来近似。

例 6-16 设计数字陷波滤波器

设计一个可去除 60Hz 电力线噪声的滤波器。抽样频率为 1000Hz 且滤波器的带宽为 4Hz。

解： 对于陷波滤波器，极点和零点位于相同的角度。本例中对应 60Hz 的角度为

$$\theta = \frac{2\pi f}{f_s} = \frac{2\pi(60)}{1000} = 0.37700\text{radians} = 21.60°$$

极点的 r 值可由式(6.34)得出。

$$带宽 \approx \frac{f_s(1-r)}{\pi}\text{Hz}$$

即

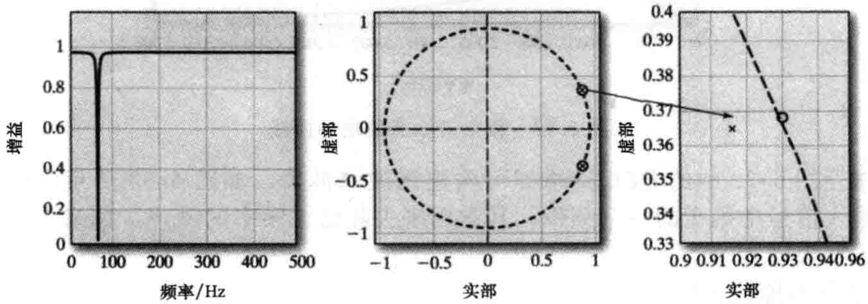
$$4 \approx \frac{1000(1-r)}{\pi}$$

求解方程中的 r 可得 $r=0.98743$

滤波器的传输函数为

$$H(z) = K\frac{z^2 - 1.8596z + 1}{z^2 - 1.8362z + 0.97502}$$

以下 MATLAB 代码可绘出图 6.34 所示的增益曲线。



a) b) c)

图 6.34 陷波滤波器的增益曲线。陷波以 60Hz 为中心且带宽不到 4Hz。在极/零点图 c 中极点在 $0.98743\angle\pm21.6°$处，而零点在 $1\angle\pm21.6°$处

```
fs = 1000;
den = [1 -1.8362 .97502];
num = [1 -1.8596 1];
[H F] = freqz(num, den, 4096, fs);
HMax = max(abs(H));
H = abs(H)/HMax;
plot(F,H);
```

6.6.3 数值直接设计——Pade 法

在 Pade 法中，首先要构建一个理想滤波器，其传输函数为 $H_D(z)$，脉冲响应为 $h_D[n]$。然后求得传输函数 $H(z)$，其脉冲响应为 $h[n]$，使得误差

$$\varepsilon = \sum_{n=0}^{\infty}(h_D[n]-h[n])^2 \tag{6.35}$$

在给定滤波器阶数的情况下最小。这就是最小平方滤波器设计。待求的传输函数形式为

$$H(z) = \frac{\sum_{n=0}^{M} b_n z^{M-n}}{z^N + \sum_{n=1}^{N} a_n z^{N-n}} \tag{6.36}$$

对应的差分方程为

$$y[n] = b_0 x[n] + b_1 x[n-1] + \cdots + b_M x[n-M] - a_1 y[n-1] - a_2 y[n-2] - \cdots - a_N y[n-N]$$

根据上式，脉冲响应为

$$h[n] = b_0 \delta[n] + b_1 \delta[n-1] + \cdots + b_M \delta[n-M] - a_1 h[n-1] - a_2 h[n-2] - \cdots - a_N h[n-N] \tag{6.37}$$

δ 函数只在其自变量为零时才有值。根据此性质，式(6.37)可改写为

$$h[n] = b_n - a_1 h[n-1] - a_2 h[n-2] - \cdots - a_N h[n-N], \quad 0 \leqslant n \leqslant M \tag{6.38}$$

及

$$h[n] = -a_1 h[n-1] - a_2 h[n-2] - \cdots - a_N h[n-N], \quad n > M \tag{6.39}$$

在式(6.38)和式(6.39)中，$h[n]$是待求未知滤波器的脉冲响应。现在需要求出 a_i 和 b_i 的值，以使得式(6.35)的误差最小。若有

$$h[n] = h_D[n]$$

可以使得误差为 0，利用该替换改写式(6.38)及式(6.39)得：

$$h[n] = b_n - a_1 h_D[n-1] - a_2 h_D[n-2] - \cdots - a_N h_D[n-N], \quad 0 \leqslant n \leqslant M \tag{6.40}$$

及

$$h[n] = -a_1 h_D[n-1] - a_2 h_D[n-2] - \cdots - a_N h_D[n-N], \quad n > M \tag{6.41}$$

在式(6.40)和式(6.41)中，存在 $N+M+1$ 个未知量(系数 a_i 和 b_i)。可在 $0 \leqslant n \leqslant M+N+1$ 范围内令 $h[n]=h_D[n]$写出 $N+M+1$ 个方程。此方法存在一个明显的局限，即在 $n>N+M+1$ 时未使 $h[n]=h_D[n]$，否则可以通过截短理想脉冲响应(矩形窗)就可以有效地实现。此方法称为 Pade 近似法(Pade approximation method)，其是满足最小平方准则的 IIR 滤波器设计。

例 6-17 Pade 近似法的应用

利用 Pade 近似法设计一个数字滤波器，所用的理想脉冲响应为

$$h[n]=\{4, 3, 2, 1, 0.5, 0.5, 0.125, 0.0625, \cdots\}$$

选 $M=N=2$。

解： 已知 $M+N+1=5$。对于本例，由式(6.40)及式(6.41)可得

$$h[n] = b_n - a_1 h_D[n-1] - a_2 h_D[n-2], \quad 0 \leqslant n \leqslant 2$$

以及

$$\begin{aligned}
h[n] &= -a_1 h_D[n-1] - a_2 h_D[n-2], \quad n > 2 \\
h[0] &= 4 = b_0 \\
h[1] &= 3 = b_1 - a_1(4) \\
h[2] &= 2 = b_2 - a_1(3) - a_2(4) \\
h[3] &= 1 = -a_1(2) - a_2(3) \\
h[4] &= 0.5 = -a_1(1) - a_2(2)
\end{aligned}$$

上述方程组以矩阵形式可表示为：

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & -4 & 0 \\ 0 & 0 & 1 & -3 & -4 \\ 0 & 0 & 0 & -2 & -3 \\ 0 & 0 & 0 & -1 & -2 \end{bmatrix} \begin{bmatrix} b_0 \\ b_1 \\ b_2 \\ a_1 \\ a_2 \end{bmatrix} = \begin{bmatrix} 4 \\ 3 \\ 2 \\ 1 \\ 0.5 \end{bmatrix}$$

求解该方程的MATLAB代码如下：

```
A = [1 0 0 0  0;...
     0 1 0 -4 0;...
     0 0 1 -3 -4;...
     0 0 0 -2 -3;...
     0 0 0 -1 -2];

c = [4;3;2;1;0.5];
b = A\c; % same as A^-1*c;
```

以上代码可以得到以下解：

$b_0=4.000$，$b_1=1.0000$，$b_2=0.5000$

$a_0=1$，$a_1=-0.5000$，$a_2=0$

对应的传输函数为

$$H(z) = \frac{4z^2 + z + 0.5}{z^2 - 0.5z}$$

利用MATLAB可得其到脉冲响应：

```
b0 = 4.0000;b1 = 1.0000;b2 = 0.5000;
a0 = 1;a1 = -0.5000;a2 = 0;
num = [b0 b1 b2];
den = [a0 a1 a2];
[h t] = impz(num, den, 10);
```

相应的脉冲响应为

$$h[n]=\{4,\ 3,\ 2,\ 1,\ 0.5,\ 0.25,\ 0.125,\ 0.0625,\ \cdots\}$$

将此脉冲响应与理想脉冲响应作比较，可以看到两者前5项完全对应，这是因为有5个方程能够完全确定这些项的系数。第5项之后，两个序列出现差异。增加阶数可以有所改善。◀

Pade近似法能够使理想滤波器与所求滤波器的脉冲响应前$N+M+1$项完全匹配。然而这种方法在实际中有一定局限，因为若要满足所需的频率指标，常需要较大的M和N，以使得$N+M+1$足够大。

在Pade近似法中，需要先利用脉冲响应的前$M+1$项得到前$M+1$个方程，因为只有这些方程中出现分子项系数(b_k)。而其余的N个方程，可以由脉冲响应中任何大于$M+1$的项确定，且不限于连续的项。以例6-17来说明，可先由$h_D[0]\sim h_D[3]$确定前4个方程，其余的可选$h_D[5]$、$h_D[7]$或$h_D[9]$。在一些情况下，恰当地选择脉冲响应的项，可以构建

更优的滤波器。

Pade 近似法有两个基本限制：首先，仅使用了理想脉冲响应的前 $N+M+1$ 个系数，而忽略了其余的系数。其次，不易估计满足设计指标的滤波器的阶数。

6.6.4 数值直接设计——Prony 法

为了解决 Pade 近似法的缺陷，可以考虑应用脉冲响应函数的统计属性，而不是脉冲响应具体的某些项。因此，建立方程组，将脉冲响应函数的自相关序列与待求滤波器的系数关联起来。该方程组称为 Prony 法。

Prony 法利用与 Pade 法相同的分子系数，可以使得设计的滤波器的脉冲响应前 M 项与待求滤波器相匹配。如果我们能够得到分母的系数，则可以利用式(6.40)计算分子系数。Prony 法在求解分母系数时，使得脉冲响应其余项的误差最小。IIR 数字滤波器的传输函数可写为

$$H(z)=\frac{\sum_{n=0}^{M}b_n z^{M-n}}{z^N+\sum_{n=1}^{N}a_n z^{N-n}}=\frac{N(z)}{D(z)}$$

对应的脉冲响应为

$$h[n]=b_n-\sum_{i=1}^{N}a_i h[n-i],\quad 0\leqslant n\leqslant M$$

以及

$$h[n]=-\sum_{i=1}^{N}a_i h[n-i],\quad n>M$$

最小化以下误差函数

$$\varepsilon=\sum_{n=0}^{\infty}(h_{\mathrm{D}}[n]-h[n])^2$$

$$E[n]=h_{\mathrm{D}}[n]-h[n]$$

作 z 变换得

$$E(z)=H_{\mathrm{D}}(z)-H(z)$$

若要最小化误差函数，可展开 ε，求其关于各个系数的偏导数，并设置这些偏导数为 0。此过程会产生出一个难以处理的非线性方程，不易得到解析解。为便于处理，重新定义误差如下：

$$E(z)=H_{\mathrm{D}}(z)-H(z)=H_{\mathrm{D}}(z)-N(z)/D(z)$$

以及

$$\hat{E}(z)=D(z)\cdot E(z)=D(z)\cdot H_{\mathrm{D}}(z)-N(z) \tag{6.42}$$

$\hat{E}(z)$有时也称为方程误差(equation error)。

z 域的乘法对应时域的卷积。在时域重写式(6.42)可得

$$D(z)\Leftrightarrow\delta[n]+\sum_{i=1}^{N}a_i\delta[n-i]$$

$$N(z)\Leftrightarrow\sum_{m=0}^{M}b_m\delta[n-m]$$

$$\hat{E}[n]=\left[\delta[n]+\sum_{i=1}^{N}a_i\delta[n-i]\right]*h_{\mathrm{D}}[n]-\sum_{m=0}^{M}b_m\delta[n-m]$$

$$\hat{E}[n]=\left[h_{\mathrm{D}}[n]+\sum_{i=1}^{N}a_i h_{\mathrm{D}}[n-i]\right]-b_n,\quad 0\leqslant n\leqslant M$$

$n>M$ 时方程可写为

$$\hat{E}[n]=\left[h_{\mathrm{D}}[n]+\sum_{i=1}^{N}a_i h_{\mathrm{D}}[n-i]\right],\quad n>M$$

当 n 远大于 M 时，平方误差为

$$\varepsilon_{\mathrm{sq}}=\sum_{n=M+1}^{\infty}\left|\left[h_{\mathrm{D}}[n]+\sum_{i=1}^{N}a_i h_{\mathrm{D}}[n-i]\right]\right|^2=\sum_{n=M+1}^{\infty}\hat{E}[n]\,\hat{E}^*[n]$$

$$\frac{\partial\varepsilon_{\mathrm{sq}}}{\partial a_j^*}=\sum_{n=M+1}^{\infty}\frac{\partial(\hat{E}[n]\cdot\hat{E}^*[n])}{\partial a_j^*}=\sum_{n=M+1}^{\infty}\hat{E}[n]\frac{\partial\hat{E}^*[n]}{\partial a_j^*}=0 \tag{6.43}$$

上式中

$$\hat{E}^*[n]=\left[h_{\mathrm{D}}^*[n]+\sum_{j=1}^{N}a_j^* h_{\mathrm{D}}^*[n-j]\right],\quad n>M$$

以及

$$\frac{\partial\hat{E}^*[n]}{\partial a_j^*}=h_{\mathrm{D}}^*[n-j]$$

将值代入式(6.43)的偏导数，可得

$$\sum_{n=M+1}^{\infty}\hat{E}[n]\frac{\partial\hat{E}^*[n]}{\partial a_j^*}=\sum_{n=M+1}^{\infty}\left[h_{\mathrm{D}}[n]+\sum_{i=1}^{N}a_i h_{\mathrm{D}}[n-i]\right]\cdot h_{\mathrm{D}}^*[n-j]=0$$

上式可写为

$$\sum_{i=1}^{N}a_i\left[\sum_{n=M+1}^{\infty}h_{\mathrm{D}}[n-i]\cdot h_{\mathrm{D}}^*[n-j]\right]=-\sum_{n=M+1}^{\infty}h_{\mathrm{D}}[n]\cdot h_{\mathrm{D}}^*[n-j],\quad j=1,2,\cdots N \tag{6.44}$$

这组复杂的方程组有 N 个方程 N 个未知量(b_1，b_2，…，b_N)。

通过定义一个与自相关函数相同的新函数，可将式(6.44)表达得更简明。

对于一个有限长 L 的序列，自相关函数定义为

$$\Phi[n]=(1/L)\sum_{i=0}^{L-1-n}x[i]x[i+n]$$

定义同样的函数如下

$$r_{\mathrm{h}}(i,j)=\sum_{n=M+1}^{\infty}h_{\mathrm{D}}[n-i]\cdot h_{\mathrm{D}}[n-j] \tag{6.45}$$

将(6.45)的定义应用到式(6.44)得

$$r_{\mathrm{h}}(0,i)=\sum_{i=1}^{N}a_i r_{\mathrm{h}}(i,j),\quad j=1,2,\cdots N \tag{6.46}$$

式(6.46)称为标准方程(normal equation)，有时也称为 Prony 标准方程。这个方程可以确定分母系数 a_1，…，a_N 的值。分子系数 b_0，…，b_M 值的求解与 Pade 法相同。

例 6-18 Prony 标准方程的应用

利用 Prony 标准方程求数字滤波器的传输函数，理想滤波器是 4 阶椭圆滤波器。

$$H(z)=\frac{0.1282z^4+0.1519z^3+0.2458z^2+0.1519z+0.1282}{z^4-1.0752z^3+1.2430z^2-0.5038z+0.1501}$$

解： Prony 标准方程在 MATLAB 中的执行形式为[num den]=prony(h，numOrder，denOrder)。这个函数中，h 是我们希望逼近的滤波器的脉冲响应，num 和 den 对应待设计滤波器 z 域传输函数的多项式，而 numOrder 和 denOrder 是希望得到的分子分母多项式的阶数。以下 MATLAB 代码完成了滤波器设计，并绘制了频率响应曲线，如图 6.35 所示。该频率响应与原 4 阶椭圆滤波器的频率响应完全相同。在 MATLAB 代码中对分子分母进行降阶会明显降低近似程度。

```
fs = 11025;
nume = [0.1282 0.1519 0.2458 0.1519 0.1282];
dene = [1 -1.0752 1.2430 -0.5038 0.1501];
[hideal T] = impz(nume, dene, 32, fs);
[num den] = prony(hideal, 4, 4);
[h T] = impz(num, den, 32, fs);
[H f] = freqz(num, den, 1024, fs);
plot(f, abs(H));
```

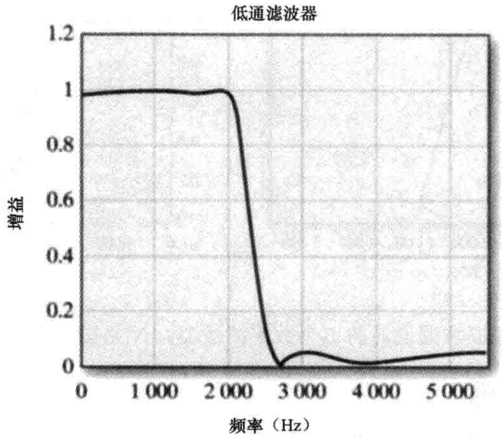


图 6.35　利用 Prony 标准方程设计低通数字滤波器

6.6.5　数值直接设计——Yule-Walker 法

在时域直接设计数字滤波器时，Yule-Walker 法与 Prony 法相似。Yule-Walker 方程为

$$r_h(k)+\sum_{i=1}^{N}a_i r_h(k-i)=\begin{cases}\sum\limits_{j=k}^{M}b_j h[j-k], & 0\leqslant k\leqslant M\\ 0, & k>M\end{cases}$$

式中，

$r_h(k)=(1/N)\sum\limits_{i=0}^{N-1-k}h[i]\cdot h[i+k]$ 是有限长序列的自相关函数。

由此可知，Yule-Walker 方程将滤波器系数 a_i 和 b_i 与自相关序列相联系。其解可由迭代法求出。若已知一个理想滤波器的频谱，则可通过傅里叶逆变换得到其脉冲响应。根据该脉冲响应，可求出其对应的自相关序列，从而根据 Yule-Walker 方程确定待设计滤波器的系数。此求解过程并不简单，MATLAB 提供了 yulewalk 函数以实现该方法。可见 Yule-Walker 法与 Prony 法类似，两者的区别在于 $r_h(k)$的定义。

例 6-19　MATLAB 中 yulewalk 函数的应用

利用 MATLAB 的 yulewalk 函数设计数字滤波器，理想滤波器如图 6.36 所示。

解： MATLAB 的 yulewalk 函数有三个参数，第一个是滤波器的阶数，其余两个是向量，其包含标准化的理想滤波器对应的 x/y 坐标。理想滤波器标准化后使 $f_s/2=1$。MATLAB 代码如下。不同阶数数字滤波器对应的幅度曲线如图 6.36。

```
fs = 11025;
fpass = 2000;
fstop = 3000;
```

```
F = [0 fpass/(fs/2) fstop/(fs/2) 1];
M = [1 1-Rpass Rstop 0];
N = 4;
[num den] = yulewalk(N, F, M);
[H f] = freqz(num, den, 1024, fs);
plot(f, (abs(H)));
```

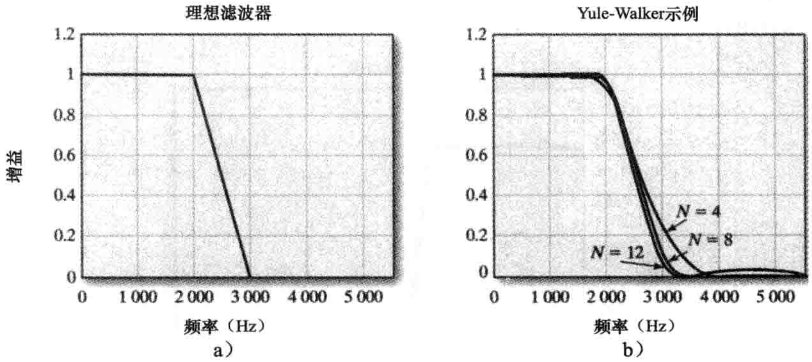


图 6.36　由图 a)中理想滤波器设计的三个数字滤波器。N 是图 b)中 yulewalk 滤波器的阶数

6.7　IIR 滤波器的应用

6.6 节讨论了谐振器和陷波滤波器，其是二阶 IIR 滤波器的应用。本节将进一步讨论 IIR 滤波器的应用，如全通滤波器、滑动平均 IIR 滤波器、梳状滤波器以及逆滤波器(inverse filter)。

6.7.1　全通滤波器

顾名思义，全通滤波器可使全部频率以单位增益通过。它的作用在于处理输入信号的相位。全通滤波器常与非线性相位的 IIR 滤波器结合使用，全通滤波器可以改善一些非线性特性引起的问题。

若一个全通滤波器为

$$H_A(z) = K\frac{N_A(z)}{D_A(z)}$$

则

$$\left|K\frac{N_A(e^{j\Omega})}{D_A(e^{j\Omega})}\right| = 1 \text{ 或 } K|N_A(e^{j\Omega})| = |N_A(e^{j\Omega})|$$

一个满足该条件的函数为

$$H_A(z) = K\frac{z-1/r}{z-r}$$

因为

$$|z-1/r| = \{[\cos(\Omega)-(1/r)]^2+\sin^2(\Omega)\}^{1/2} = (1/r)[r^2-2r\cos(\Omega)+1]^{1/2}$$

以及

$$|z-r| = \{[\cos(\Omega)-r]^2+\sin^2(\Omega)\}^{1/2} = [r^2-2r\cos(\Omega)+1]^{1/2}$$

所以有

$$\left|\frac{z-1/r}{z-r}\right| = 1/r$$

若选 $K=r$，则在任何频率下，幅频响应$|H_A(e^{j\Omega})|=1$。

一般地，全通滤波器可由如下传输函数表示

$$H_A(z) = K\cdot\prod_{i=1}^{L}\frac{z-(1/r_i)}{z-r_i} \tag{6.47}$$

上式中，r_i 可以是实数也可以是复数。若 r_i 是复数，为使滤波器系数为实数，则其共轭复数也应包含在此传输函数之中。增益常量 K 使滤波器在整个频带中的增益归一化。图 6.37给出不同 r_i 值所对应的滤波器相移(phase shift)变化。

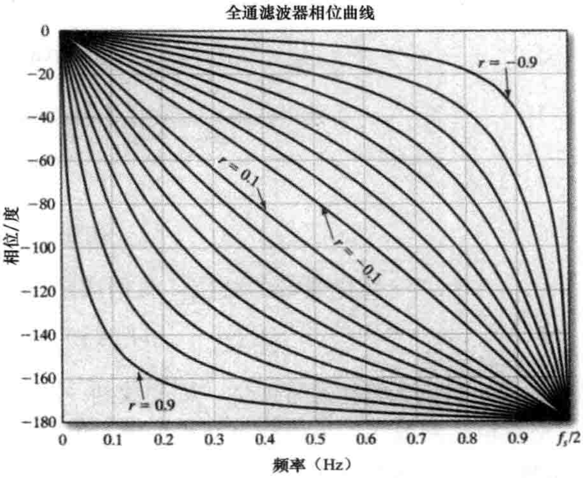


图 6.37 含一个实极点和一个实零点的全通滤波器的相移，极点以 0.1 的步长从−0.9 移动到+0.9

虽然全通滤波器不能提供一个明确的方法来修正 IIR 滤波器的非线性相位，但它提供了一种机制，可使部分频带的相位得到修正。

例 6-20 利用全通滤波器实现线性化

设计一个满足以下指标的切比雪夫低通滤波器。

抽样频率：44.1kHz

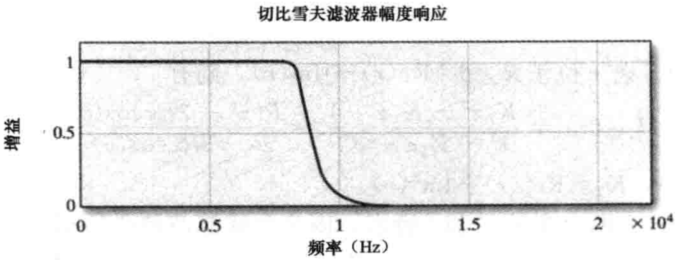
通带：0Hz→8kHz，波动低于 0.001

阻带：15kHz→$f_s/2$，波动低于 0.0001

试说明通过一阶全通滤波器，上述滤波器的相位曲线在通带部分可以线性化。取直线与相位曲线间的最大偏差作为线性的度量。迭代以确定全通滤波器参数 r 的最佳取值，该全通滤波器的传输函数类似式(6.47)表示的传输函数。

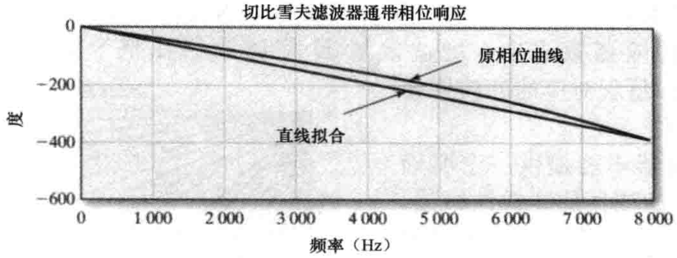
解： 先利用 MATLAB 计算并画出滤波器的频率响应。幅度和相位曲线如图 6.38 所示。相位曲线的通带部分加了一条直线对比其非线性。

```
fs = 44100;                                %Sample frequency
Rp = .001;RpDB = -20*log10(1-Rp);          %Passband ripple
Rs = .0001;RsDB = -20*log10(Rs);           %Stopband ripple
fpass = 8000;fstop = 15000;
%Get the order for a Chebyshev type 1 filter.
[N Wn] = cheb1ord(fpass/(fs/2),fstop/(fs/2),RpDB,RsDB);
[num den] = cheby1(N,RpDB,Wn);             %Create a filter of that order
[Hc f] = freqz(num,den, 512, fs);          %Find the frequency response
figure(1); clf;
subplot(2,1,1)
plot(f, abs(Hc))                           %Plot the magnitude
subplot(2,1,2)
HcAngle = unwrap(angle(Hc))*180/pi;        %Put the angle in degrees
plot(f, HcAngle)                           %plot the phase curve
line([0 f(i)],[0 HcAngle(i)]);             %Draw a line from 0 to passband edge
```



a)

图 6.38 切比雪夫 I 型低通滤波器的幅度 a 和相位 b 曲线，直线用来对比相位曲线的非线性



b）

图 6.38 （续）

为了线性化通带的相位，我们应用一个一阶全通滤波器，传输函数为

$$H_{\mathrm{All}}(z)=r\frac{z-1/r}{z-r}$$

欲使相位曲线在通带线性，则要找到一个 r 值，使通带范围内相位曲线与直线之差的最大绝对值达到最小。尽管找不到闭形解，但可以通过 MATLAB 迭代连续改变的 r 值以得到“最优”解。比较图 6.38 中的相位曲线与图 6.37 中的相位曲线可知，我们需要一个正 r 值进行修正。在 MATLAB 中，可设定一个循环，从 $r=0.1$ 开始，以 0.001 的增量进行迭代。在每次迭代中，都会得到全通滤波器新的传输函数，乘以原切比雪夫滤波器的传输函数，得到相乘之后传输函数的频率响应。比较此相位响应与直线之间误差的最大绝对值，不断迭代 r 值，直到二者差的最大绝对值不再减小后停止迭代。通过 MATLAB 得到的最后结果为 $r=0.7990$，修正后的相位曲线与直线的最大差值为 18.2°。结果如图 6.39 所示。

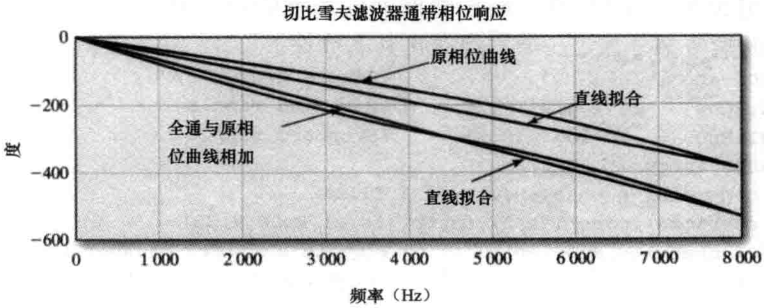


图 6.39 通带范围内，原切比雪夫滤波器以及其与全通结合的滤波器的相位曲线，两种情形下都通过阻带边缘的直线作比较

若要从全通滤波器得到更多的修正，可根据式(6.47)增加多个级联的一阶全通滤波器。若 r 的值是复数，则二阶全通滤波器的传输函数为

$$H_{2\mathrm{All}}(z)=\frac{z-1/r}{z-r}\cdot\frac{z-(1/r)^*}{z-r^*}$$

式中，$*$ 表示共轭。若 r 位于 $R\angle\theta=\mathrm{Re}(r)+\mathrm{jIm}(r)$，则有

$$H_{2\mathrm{All}}(z)=\frac{K_2z^2-K_1z+1}{z^2-K_1z+K_2}=\frac{R^2z^2-2Rz\cos(\theta)+1}{z^2-2Rz\cos(\theta)+R^2}\tag{6.48}$$

式中，$K_1=2\mathrm{Re}(r)$，$K_2=\mathrm{Re}^2(r)+\mathrm{Im}^2(r)$。

以极坐标的形式有两个变量 R 和 θ 待求。因为全通滤波器必须稳定，所以有 $0<R<1$，又因同时包含极点及其共轭点，则只需 $0\leqslant\theta<180°$。由此可在单位圆的上半部分迭代 R 和 θ 求出最优值，使通带范围修正后的相位曲线与直线的最大绝对差值最小。对例 6-20 的切比雪夫低通滤波器应用二阶全通滤波器，迭代后求最优解得，$R=0.703$，$\theta=23.60°$，而

最大差值只有 7.89°。

6.7.2　IIR 滑动平均滤波器

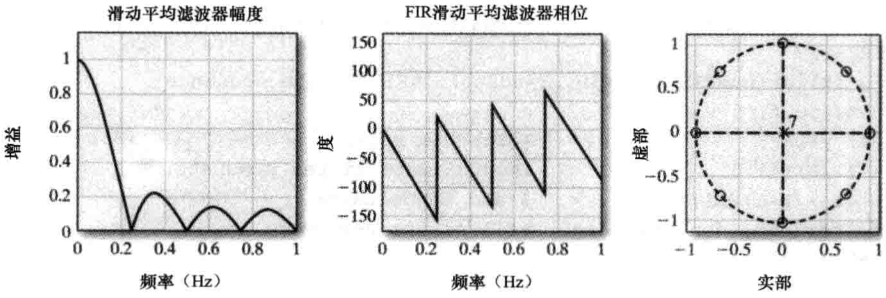
为了计算长度 L 的序列 $x[k]$的滑动平均，可利用下列差分方程：

$$A_{\mathrm{M}}[k]=\frac{1}{L}\sum_{i=k-L+1}^{k}x[i] \tag{6.49}$$

式中，$A_{\mathrm{M}}[k]$是 k 时刻的滑动平均值，x 是输入序列，L 是输入序列长度。由 z 变换可见，此实质上为 FIR 滤波器方程。

$$H_{\mathrm{M}}[z]=\frac{z^{L-1}+z^{L-2}+\cdots+1}{Lz^{L-1}} \tag{6.50}$$

图 6.40 给出了长度为 8 的 FIR 滑动平均滤波器的幅度、相位及 z 平面图。特别注意，幅度曲线与矩形窗的曲线相同。可以推断，该数字滤波器的脉冲响应在时域为矩形序列。



a）幅度曲线　　b）相位曲线　　c）极/零点图

图 6.40　7 阶 FIR 滑动平均滤波器(长度为 8)

通过一些处理，式(6.50)可改写为 IIR 滤波器的方程。展开式(6.49)得

$$A_{\mathrm{M}}[k]=\frac{1}{L}[x[k]+x[k-1]+x[k-2]+\cdots+x[k-L+2]+x[k-L+1]] \tag{6.51}$$

同样

$$A_{\mathrm{M}}[k-1]=\frac{1}{L}[x[k-1]+x[k-2]+x[k-3]+\cdots+x[k-L+1]+x[k-L]] \tag{6.52}$$

式(6.51)减去式(6.52)得

$$A_{\mathrm{M}}[k]=A_{\mathrm{M}}[k-1]+\frac{1}{L}[x[k]-x[k-L]] \tag{6.53}$$

式(6.53)是由非递归方程推导出的递归关系式。相应传输函数为

$$H_{\mathrm{M}}[z]=\frac{1}{L}\frac{z^{L}-1}{z^{L-1}(z-1)} \tag{6.54}$$

可以看到，式(6.54)与式(6.50)相比较，除了在分子分母都多了一项$(z-1)$外，其余都相同——即在 $z=+1$ 处，分别多了一个极点和一个零点。

因为同样有 $L-1$ 项，所以从计算上 IIR 滑动平均滤波器并不比 FIR 优秀，同时式(6.54)还有一些附加项，但并不增加计算量。然而，通过 IIR 滑动平均滤波器近似一个滑动平均数，可以提高一定的效率。式(6.53)中的项 $x[k-L]$可由均值 $A_{\mathrm{M}}[k-1]$近似。则差分方程可简化为

$$A_{\mathrm{M}}[k]=\frac{L-1}{L}A_{\mathrm{M}}[k-1]+\frac{1}{L}x[k] \tag{6.55}$$

相应传输函数为

$$H_M[z]=\frac{1}{L}\cdot\frac{z}{z-(L-1)/L} \tag{6.56}$$

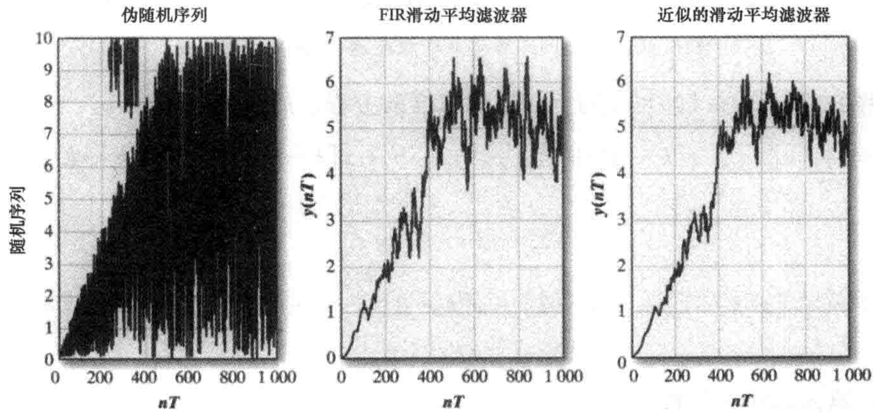
例 6-21　MATLAB 中 filter 函数的应用

以下 MATLAB 代码生成了 1000 个随机整数，其中，前 500 个数的均值位于 0～5 之间，后 500 个数均值为 5。先用长度为 20 的 FIR 滑动平均滤波器对这一序列进行滤波，再用式(6.56)的近似滑动平均滤波器对该序列滤波，长度同样为 20，比较二者的结果。

```
randSeq = (1:500)/500;
a = 0; b = 10;
randSeq = randSeq.*(a + (b-a).*rand(1, 500));
randSeq = [randSeq (a + (b-a).*rand(1, 500))];
```

解： MATLAB 代码如下所示，应用 filter 函数完成题目要求。结果如图 6.41 所示。

```
L = 20;                        %Length
nT = [1:1000];                 %x-axis vector
numFIR = (1/L)*ones(1, L);     %Create an FIR filter of length L
denFIR = 1;
yFIR = filter(numFIR, denFIR, randSeq); %Filter the sequence
figure(1);clf;
plot(f, yFIR);                 %Plot the result
num = (1/L)*[1 0];             %Approximate filter numerator
den = [1 -(L-1)/L];            % and denominator
y = filter(num, den, randSeq); %filter the sequence
figure(2);clf;
plot(f, y, 'k');               %Plot the result
```



a）伪随机序列　　b）经长度为20的FIR滑动平均滤波器滤波后的序列　　c）伪随机序列经过近似滑动平均滤波器滤波，传输函数见式（6.56）

图 6.41　例 6-21 图 ◀

6.7.3　IIR 梳状滤波器

此类滤波器之所以称为梳状滤波器，是因为其幅度响应类似一个齿状向上的梳子。梳状滤波器可用来分离出按照周期性间隔出现的数据。可能用到梳状滤波器的实例有：(1)观测太阳、月亮、潮汐等周期性出现的现象。(2)雷达或声呐图像，其追踪一个移动目标以及每幅图中背景出现的位置。(3)滤除周期性噪声(如电力线噪声及其谐波)。

与滑动平均滤波器相同，梳状滤波器可由 FIR 滤波器实现或 IIR 滤波器实现。在 FIR 滑动平均滤波器 $z=+1$ 处添加一个零点，即可实现 FIR 梳状滤波器。其传输函数为

$$H_C(z)=K\frac{z^N-1}{z^N} \tag{6.57}$$

图 6.42 所示为一个典型的 10 阶 FIR 梳状滤波器。

将极点从原点放射地移向单位圆附近，并与零点以一定间隔排列，则会使梳状滤波器的波瓣锐化。这样就变成了 IIR 数字滤波器，同时也会增加计算量。这类滤波器通过放置零极点很容易构建。“锐化”滤波器的传输函数为

$$H_{CS}(z) = K\frac{z^N - 1}{z^N + r^N} \tag{6.58}$$

式中，r 是极点的模。图 6.43 给出了 10 阶锐化梳状滤波器的零极点图和幅度曲线。

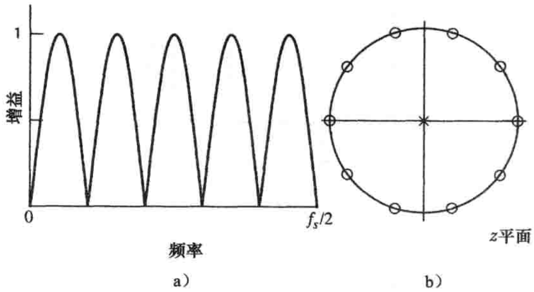


图 6.42　10 阶 FIR 梳状滤波器，通过向滑动平均滤波器添加零点得到。若阶数为奇数，则在 $z=-1$处没有零点且 $f_S/2$ 处增益为 1

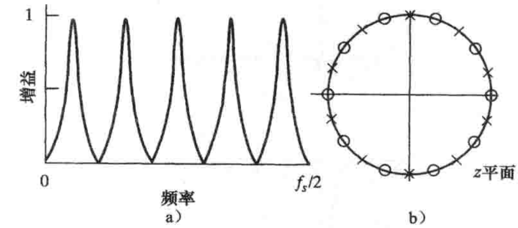
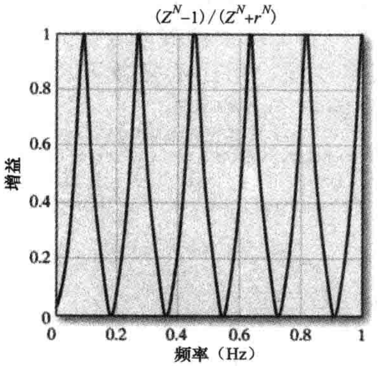
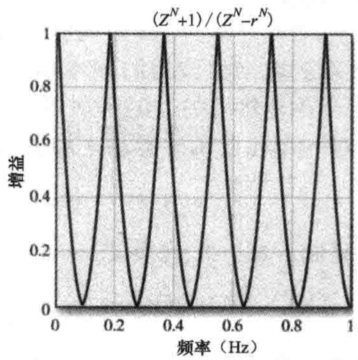
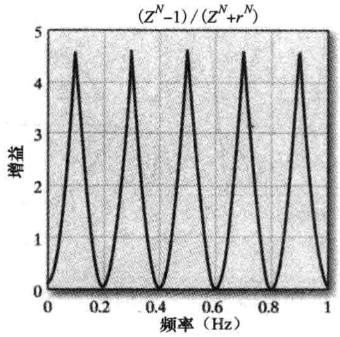
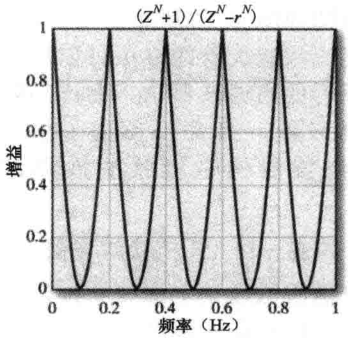


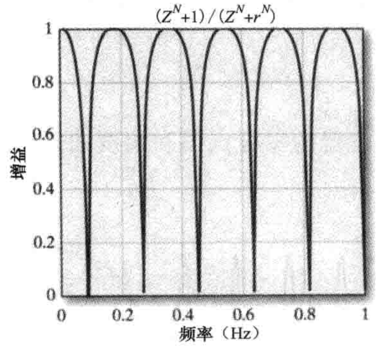
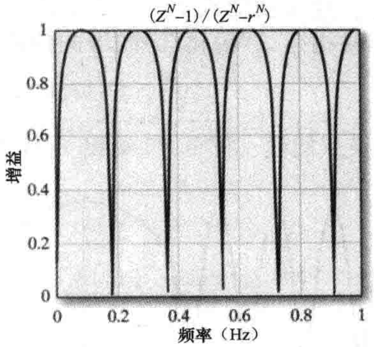
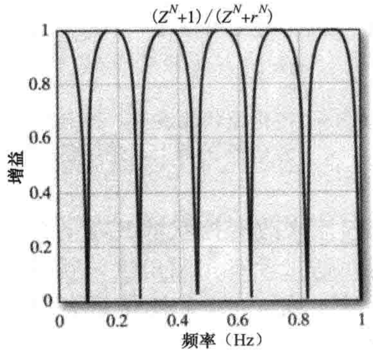
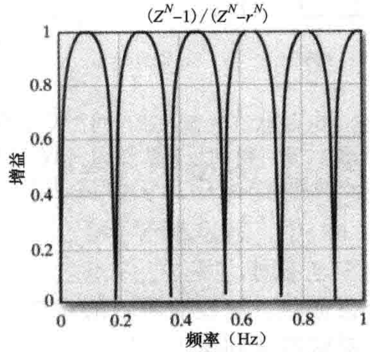
图 6.43　锐化梳状滤波器，极点位于半径为 0.95 的圆上，并与零点等间隔

在式(6.58)中，通过改变零极点的符号及阶数的奇偶性来改变梳状滤波器的形状。若极点符号与零点不同，则位于零点之间；若与零点相同，则二者有相同角度，且滤波器有更典型的梳齿形状。滤波器阶数的奇偶性决定了在 $z=+1$ 或 $z=-1$ 处是否有极点或零点。表 6.5 对这些区别做了说明。

表 6.5　(a)极/零点符号不同的 IIR 梳状滤波器。(b)极/零点符号相同的 IIR 梳状滤波器

	(a)极点和零点符号不同	
偶数阶		
奇数阶		

（续）

<table>
<tr><td></td><td colspan="2">(b)极点和零点符号相同</td></tr>
<tr><td>偶数阶</td><td>
</td><td>
</td></tr>
<tr><td>奇数阶</td><td>
</td><td>
</td></tr>
</table>

设计梳状滤波器的第二种方法涉及映射函数。假如我们选取一个在0Hz处只有一个主波瓣的滑动平均滤波器(其通带边界频率为 f_z)，然后作代换 $z^R \to z$，就可得到一个新滤波器，其在 $f_z/2 \leqslant f \leqslant f_S/2$ 区间将主波瓣重复了 R 次。为了便于理解此过程，我们通过代换 $e^{j\omega T} \to z$ 得到数字滤波器的频率响应。为了展缩频率变量，用比例常数 R 乘以 ω。因为 $e^{j\omega RT}$ 对应 z^R，显然增加 z 的幂次等价于对频率进行展缩。

下面通过一个4阶IIR滑动平均滤波器来说明该方法：

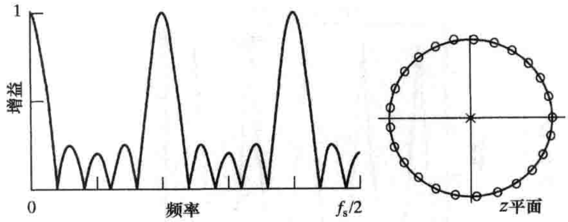
$$H_M(z) = \frac{1}{5}\frac{z^5-1}{z^4(z-1)}$$

将该滤波器复制5次，作代换 $z^5 \to z$ 得

$$H_C(z) = K\frac{z^{25}-1}{z^{20}(z^5-1)} = K\frac{z^{25}-1}{z^{25}-z^{20}}$$

式中，K 用来使得最大增益归一化。幅度曲线和零极点图如图6.44所示。

这种构建梳状滤波器的方法可以应用到任何窄频带的低通滤波器中，而不局限于滑动平均滤波器。例6-22阐述了由一个三阶低通滤波器构造梳妆滤波器的过程。



a)　　b)

图6.44　在频率空间复制滑动平均滤波器所得到的梳状滤波器

例6-22　构建一个IIR梳状滤波器

利用三阶椭圆低通数字滤波器构建一个重复因子为7的梳状滤波器。

$$H_E(z) = 0.021714\,\frac{z^3 - 0.848137z^2 - 0.848137z + 1}{z^3 - 2.7117z^2 + 2.4905z - 0.77213}$$

滤波器通带边界在 30Hz 处，阻带边界在 60Hz 处。通带和阻带的波动为 0.05，抽样频率为 1000Hz。

解：三阶椭圆低通数字滤波器的幅度曲线如图 6.45 所示，其为一个窄带滤波器。因为重复因子 R=7，故对传输函数做代换 $z^7 \to z$。

梳状滤波器的传输函数为

$$H_C(z) = 0.021714\,\frac{z^{21} - 0.848137z^{14} - 0.848137z^7 + 1}{z^{21} - 2.7117z^{14} + 2.4905z^7 - 0.77213}$$

图 6.46 给出了此梳状滤波器的幅度曲线和零极点分布图。

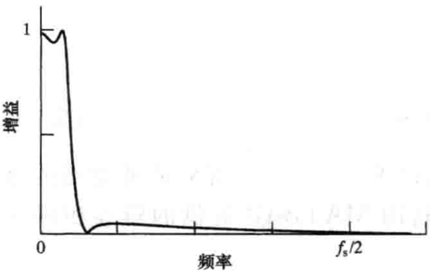
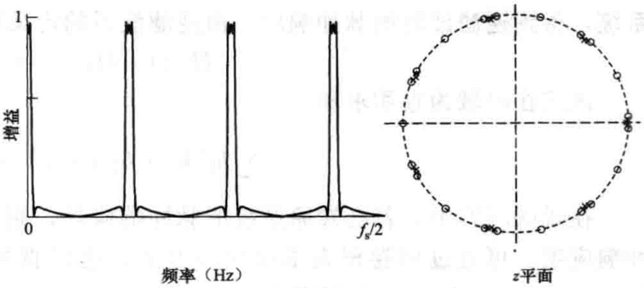


图 6.45　三阶低通椭圆数字滤波器



a)　　b)

图 6.46　梳状滤波器的幅度曲线和极/零点图，由三阶椭圆低通滤波器构建，重复因子为 7　◀

6.7.4　逆滤波器

考虑这种应用情形，在数据传输过程中，传输线(transmission line)改变了数据的频率特性，而这种改变造成了数据失真。在多数情况下，有可能通过滤波器对数据进行处理以修正传输线造成的失真，该滤波器具有与传输线系统相反的频率响应。这种滤波器称为逆滤波器。

若一个离散系统的传输函数为 $H(z)=N(z)/D(z)$，则该离散系统的逆滤波器为 $H_I(z)=D(z)/N(z)$，从而可使得 $H(z)H_I(z)=1$。构建逆滤波器显然存在一个问题，即原系统的零点将成为逆滤波器的极点，因此一个在单位圆上或单位圆外有零点的系统将产生一个不稳定的逆滤波器。

例 6-23　为传输线构建一个逆滤波器

现有某传输线，其在源端和负载端的阻抗都不匹配。这种情况下，传输的信号在两端都向传输信道反射。若该传输信道的脉冲响应为

$$h[n] = [1, 1/4, 1/16, 1/64, \cdots]$$

求其对应的逆滤波器。

解：本例中，可将脉冲响应以闭合形式表示

$$h[n] = (1/4)^n$$

相应的 z 变换为

$$(1/4)^n \Leftrightarrow \frac{z}{z - 1/4} = H(z)$$

其逆滤波器为

$$H_I(z) = \frac{z - 1/4}{z}$$

该逆滤波器是FIR滤波器，其脉冲响应为

$$h_{\mathrm{I}}[n]=\{1,-1/4,0,0,\cdots\}$$

若将此逆滤波器级联到存在反射的传输线上，修正后的系统传输函数为

$$H_{\mathrm{C}}(z)=\frac{z}{z-1/4}\frac{z-1/4}{z}=1$$

或在时域中，修正后系统的脉冲响应由传输线的脉冲响应与逆滤波器的脉冲响应卷积得到。

$$h_{\mathrm{C}}[n]=h[n]*h_{\mathrm{I}}[n]=\sum_{k=0}^{\infty}h[k]\cdot h_{\mathrm{I}}[n-k]=\delta[n]$$

在某些应用中，无法将系统函数 $H(z)$ 以闭合形式表示。此时，可以在时域求出这些系统，得到逆滤波器的脉冲响应。由逆滤波器的定义可得

$$H(z)\cdot H_{\mathrm{I}}(z)=1$$

该式在时域为卷积求和。

$$\sum_{k=0}^{\infty}h_{\mathrm{I}}[k]\cdot h[n-k]=\delta[n] \tag{6.59}$$

在式(6.59)中，若已知原系统的脉冲响应项，则通过求解 $h_{\mathrm{I}}[n]$ 可得到逆滤波器的脉冲响应项。可通过列卷积表求解(6.59)式，也可直接利用MATLAB提供的解卷积函数deconv(deconvolution)进行求解。

例 6-24 构建一个传感器响应的逆滤波器

一个传感器的冲激响应函数为

$$h(t)=\left|\mathrm{e}^{-t/0.01}\sin(200\pi t)\right|$$

利用式(6.59)求解其逆滤波器的脉冲响应。

解： 传感器的冲激信号为模拟信号，需要对其抽样得到对应的脉冲响应。

以下MATLAB代码对冲激响应进行抽样，得到对应的脉冲响应。

```
Tau = .01;f = 100;
t = linspace(0,.8,64);
for i = 1:512
h(i) = abs(exp(-t(i)/Tau)*sin(2*pi*f*t(i)));
end
```

将 $h[n]$ 的值代入到式(6.59)可解出逆滤波器的脉冲响应 $h_{\mathrm{I}}[n]$。要做到这点，需要反向求解卷积方程。MATLAB提供的函数deconv可用来求解反卷积(解卷)。因此，需要从脉冲响应中反卷积出 δ 函数。若把脉冲响应的系数看作多项式系数，则两个脉冲响应方程的卷积相当于两个多项式相乘。反卷积相当于两个多项式相除。MATLAB中deconv函数调用形式为[q r]=deconv(a，b)。

若 a 和 b 是表示多项式系数的向量，则 $a/b=q$ 余数为 r。本例中，需要一个足够长的向量来表示 δ 函数。在MATLAB中可由以下语句完成：

```
delta = zeros(1, 127);
delta(1) = 1;
```

逆滤波器的脉冲响应项可由deconv函数得到。

```
[hinv r] = deconv(delta,h);
```

滤波器的传输函数可以表示为FIR滤波器，或利用Prony法表示为IIR滤波器。不论哪种情况，在时域进行解卷时，脉冲响应都需要被截为有限长，因此解卷的结果是一个近似值。对于IIR滤波器，10阶的滤波器可得到一个适当的结果。

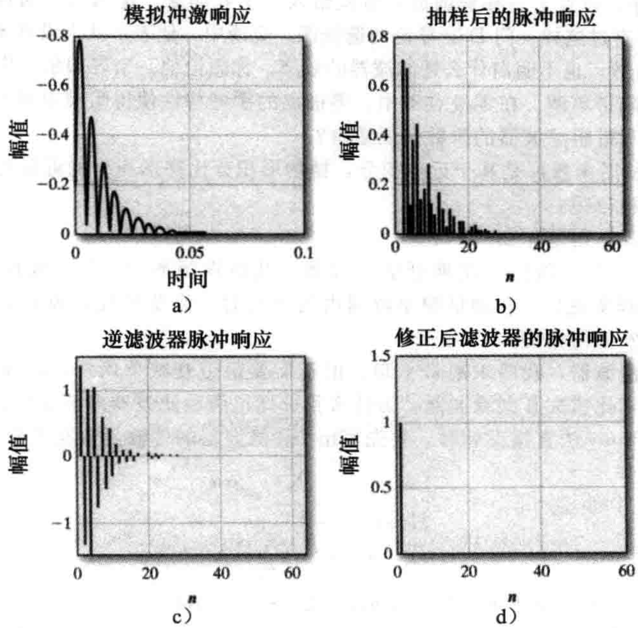


图 6.47 本图给出了用 MATLAB 的 deconv 函数生成逆滤波器的脉冲响应项，修正后的脉冲响应是原滤波器及其逆滤波器级联后的脉冲响应

小结

本章讨论了 IIR 滤波器的设计和分析方法。利用双线性变换法进行 s 平面到 z 平面的映射，从而将经典的模拟巴特沃斯、切比雪夫和椭圆滤波器转化为数字滤波器。三种模拟滤波器的不同特性为我们提供了选择的余地，类似于 FIR 滤波器设计中提供了不同窗函数。也可以直接构建 IIR 滤波器，即根据滤波器的设计指标，直接得到其 z 域传输函数，中间过程无须构建模拟滤波器。通过一个简单的计算器，我们就可构建满足指标的陷波滤波器和谐振器。更复杂的直接设计法有 Pade 法、Prony 法和 Yule-Walker 法。

本章最后以一些应用结束，包括全通滤波器、滑动平均滤波器、梳状滤波器和逆滤波器。

习题

概念题

1. 全通滤波器在整个频带上的增益都为 1，它有何作用？
2. 对于一个巴特沃斯低通滤波器，当抽样频率增大而过渡带边界频率保持不变时，极点和零点有什么变化？例如，零点会向极点移动吗？极点会移动吗？等等。
3. 设计一个带阻滤波器，阻带在 1300Hz 处且带宽为 100Hz。要求阻带波动必须为 0.01 或更小，通带波动可以为 0.1。有人建议你使用脉冲响应不变法来设计，这是一个好主意吗？
4. 在直接设计 IIR 滤波器时，Prony 法和 Yule-Walker 法比 Pade 法的优点在哪里？
5. 对称的分子系数在实现过程中有何好处？
6. 当使用代码(如 C++)实现数字滤波器时，试说明什么决定了实际的抽样频率。
7. 假设给你一个通过汉明窗设计的 FIR 滤波器的传输函数。若不需要线性相位，请详细说明如何使用 IIR 滤波器技术重新设计此滤波器。
8. 若在单位圆上放置一个零点，利用 z 平面上极点和零点的向量，说明数字滤波器相位曲线的变化。
9. 现在是 2032 年，你已经离校 20 年。因为传送器入口的故障，你被困在了 Tobongobongo 岛[⊖]，而唯一的去处是入口附近的一个酒吧。你进去后发现吧台后面是一个漂亮的机器人，于是你在吧台边坐下，

⊖ Tobongobongo 是 Wiley Miller 的漫画 Non sequitur 中虚构的岛屿。作者曾幻想写完本章习题时到那里去。

将折角的DSP书放好，并点了一杯葡萄酒，希望给人一个好印象。递酒时，酒保发现了你的DSP书，并提到他在酒保大学有过这样一门DSP导论的选修课。交谈中，他表示从来没真正明白为什么IIR滤波器比FIR滤波器更有效，也不明白什么是滤波器的效率。你想给他一个好印象，你怎么解释这些问题？

10. 一些低通滤波器的阻带单调。在零极点图中，零极点的哪些特性使得阻带单调？
11. 为什么大多数带通和带阻滤波器的阶数是偶数阶？
12. 一些 s 域到 z 域的映射函数都是基于近似积分，抛物形积分比梯形积分的近似更精确。但为什么没有广泛地应用抛物形积分呢？
13. 什么是频率变换？其作用是什么？
14. 假设你已经设计出一个8阶巴特沃斯低通滤波器，其抽样频率600Hz，截止频率125Hz，当你在ARM7处理器中编程实现后，在测量频率范围内的增益时，你发现截止频率实际为140Hz。最可能的误差原因是什么？如何解决？
15. 假设我正设计一个滤波器，我要求通带平坦，但不需要阻带衰减单调。若线性相位也不需要，则可考虑巴特沃斯或者切比雪夫Ⅱ型滤波器。为什么我选择这两种滤波器而不是其他滤波器？
16. 在利用Pade法或Prony法直接设计时，要先写出IIR滤波器的传输函数及差分方程如下：

$$H(z)=\frac{\sum_{k=0}^{M} b_k z^{M-k}}{z^N+\sum_{k=1}^{N} a_k z^{N-k}}$$

$$h(k)=b_k-a_1h(k-1)-a_2h(k-2)-\cdots-a_Nh(k-N),\quad 0\leqslant k\leqslant M$$

$$及\ h(k)=-a_1h(k-1)-a_2h(k-2)-\cdots-a_Nh(k-n),\quad k>M$$

说明这两种方法的异同点。

17. 椭圆滤波器被认为是一种等波动滤波器，怎么理解其含义？

分析和设计题

6.1 节

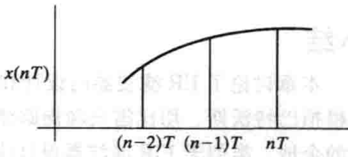


图 P6.1 利用抛物线近似积分

6.1 双线性变换由梯形积分推导得来。Simpson法则提出一种近似积分的方法，用抛物线代替梯形。图P6.1为一段抛物线，从 $(n-2)T$ 到 nT 的面积为

$$面积=\frac{2T}{3}[x(n-2)T+4x(n-1)T+x(nT)]$$

(a) 利用上式推导在 z 平面的"抛物线变换"表达式。

(b) 对于二阶或更高阶的曲线，显然抛物线积分比梯形积分得到更精确的近似值，但为什么不用？

6.2 若某 s 域传输函数，其分母为 N 阶，分子为 M 阶。经过双线性变换后，试证明其对应的 z 域传输函数在 $z=-1$ 处至少有 $N-M$ 个极点。

6.3 现有 s 平面的巴特沃斯滤波器经过双线性变换映射到 z 平面，若抽样频率为1000Hz且没有进行预畸变，以下 s 平面的频率对应在 z 平面的频率为多少？

s 平面的频率→0Hz，100Hz，250Hz，400Hz，500Hz。

6.4 一个具有低通滤波器频率特性的 RC 电路，其传输函数为

$$H(s)=\frac{1/(RC)}{s+1/(RC)}$$

(a) 若该滤波器的截止频率为 $f_c=1/(2\pi RC)=320\text{Hz}$，则合适的抽样频率应为多少？说明理由。

(b) 若将截止频率改为 f_c'，f_c' 为预畸变后的频率，抽样频率设定为2000Hz，求解新的传输函数 $H(s)$。

(c) 对预畸变后的滤波器 $H(s)$ 进行双线性变换，并用MATLAB在同一坐标系下画出 $H(s)$ 和 $H(z)$ 的幅度响应。验证在 f_c 处满足 $|H(z)|=|H(s)|$。

(d) 重复(c)，但要用到式(6.5)。选择式中常数使得 z 域和 s 域的幅度曲线在400Hz处相等。

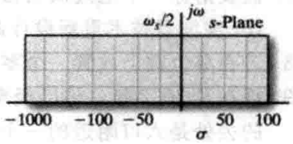


图 P6.5 s 平面

6.5 利用MATLAB构建一个基于双线性变换的 s 平面到 z 平面的映射，s 随 s 平面的 ω 线和 σ 线变化。只进行图P6.5中 s 区域的映射并添加适当标记。选取抽样频率为 $f_s=1\text{kHz}$。

6.6 利用梯形近似曲线下面积并列出积分的差分方程可以推导出双线性变换。同样可以利用某一点 kT 到 $(k-1)T$ 的直线斜率近似得到函数在 kT 点的导数，如图 P6.6 所示。

在 kT 点的导数约为

$$\frac{\mathrm{d}y}{\mathrm{d}t} \approx \frac{y(kT) - y[(k-1)T]}{T}$$

在拉普拉斯变换域中乘以 s 相当于时域的微分。利用此方法，给出一个新的从 s 域到 z 域的映射函数。该函数称为反向差分法(method of backward differences)。

6.7 利用习题 6.6 的结果，画出 s 域草图，并说明利用上述映射函数，s 左半平面和 $j\omega$ 轴是如何映射到 z 平面。分析此映射函数是否可以保持系统的稳定性？

6.8 双线性变换的基础是利用梯形近似估计曲线下面积。同样也可以用矩形近似曲线下面积，如图 P6.8 所示。试证明利用矩形积分得到的映射函数与习题 6.6 中反向差分法得到的结果相同。

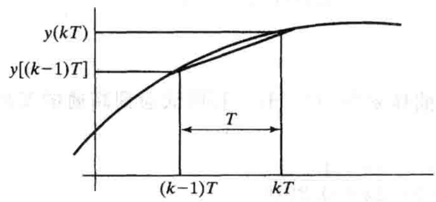


图 P6.6 利用两个相邻抽样点间的直线斜率近似导数

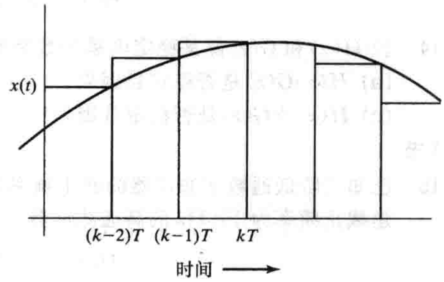


图 P6.8 利用矩形近似曲线下面积

6.9 一个模拟滤波器的截止频率为 ω_c。利用双线性变换构造一个数字滤波器，该截止频率经过预畸变之后的频率记为 $\omega_{prewarp}$。当抽样频率 f_s 变得非常大时，ω_c 与 $\omega_{prewarp}$ 会如何变化？二者会靠近还是分开？说明理由。

6.10 以下关于双线性变换的命题哪些为真？

(a) 稳定的模拟滤波器生成稳定的数字滤波器。

(b) 将 s 平面无限的频率空间映射到 z 平面有限的频率空间。

(c) 不能应用于非带限的模拟滤波器。

(d) 可根据梯形积分推导得到。

(e) 以上全是。

(f) 以上全不是。

6.11 对以下模拟滤波器应用双线性变换：

$$H(s) = \frac{s+1}{s^2 - 2s + 5}$$

回答下列问题：

(a) 在 $z=-1$ 处有多少个极点？

(b) 在 z 域此滤波器稳定吗？解释原因。

(c) z 域数字滤波器的分子与分母的阶数是多少？

(d) 若抽样频率为 50Hz，s 平面 10Hz 的频率对应于 z 平面是多少？

6.2 节

6.12 现有如下系统

$$H(z) = \frac{K(z^2 + 2z + 1)}{z^2 - r\cos(\pi/4)z + r^2}$$

在 $z=-1$ 处有两个零点，复极点位于 $r\angle\pm(\pi/4)$ 处。利用 MATLAB 分别画出该低通滤波器在 $K=0.2688$，$r=0.8$ 以及 $K=0.4197$，$r=1.25$ 时的幅度曲线。为什么这两个系统有相同的幅度曲线？其中之一是否稳定？解释原因。

6.13 系统如图 P6.13 所示，求使系统稳定的 K_1 和 K_2 值的范围。

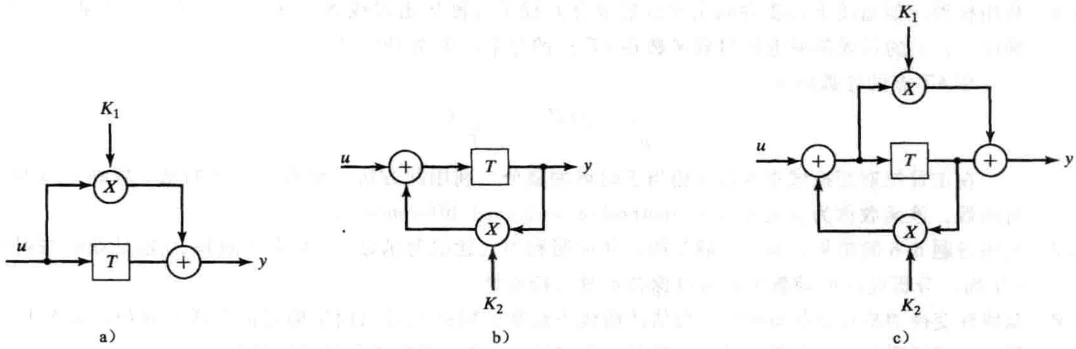


图 P6.13　确定使系统稳定的常数范围

6.14　若 $H(z)$ 和 $G(z)$ 都是稳定因果的数字滤波器，回答下列问题：

(a) $H(z)G(z)$ 是否稳定且因果？　(b) $H(z)/G(z)$ 是否稳定且因果？

(c) $H(z)+G(z)$ 是否稳定且因果？

6.3 节

6.15　已知二阶低通数字滤波器的截止频率为 150Hz，抽样频率 1000Hz。利用低通到高通的变换方程构建截止频率为 150Hz 的高通滤波器。

$$H(z)=0.1275\frac{z^2+2z+1}{z^2-0.7818z+0.2919}$$

6.4 节

6.16　在巴特沃斯、切比雪夫Ⅰ型、切比雪夫Ⅱ型及椭圆滤波器这 4 种滤波器中，哪些具有以下性质？

(a) 平坦的通带(低通滤波器)。　(b) 单调的阻带(低通滤波器)。

(c) 给定阶数时，最窄的过渡带。　(d) 全极点模拟滤波器。

(e) 所有频带中 RMS 误差最小。

6.17　下列哪些性质适用于 IIR 滤波器？

(a) 有多重反馈项。　(b) 在原点处可能没有极点。

(c) 可能不稳定。　(d) 一般具有线性相位。

(e) 脉冲响应项全为正。　(f) 以上全是。

(g) 以上全不是。

6.18　下列哪些性质适用于椭圆滤波器？

(a) 通带平坦。

(b) 原点处有极点。

(c) 满足相同指标时，一般比巴特沃斯滤波器的阶数低。

(d) 阻带单调。

(e) 差分方程中有反馈项。

(f) 以上全是。

(g) 以上都不是。

6.19　设计一组(共 4 个)满足以下指标的数字滤波器。抽样频率为 44.1kHz。4 个滤波器分别为巴特沃斯、切比雪夫Ⅰ型、切比雪夫Ⅱ型及椭圆滤波器。每种情况需求出满足指标的最低阶。利用 MATLAB 画出各滤波器的幅度曲线，验证你的设计满足指标要求。曲线放在同一个图中，放大显示通带、过渡带及阻带。标出每个滤波器的阶数。

类别	低通
通带	0～4500Hz
通带波动	0.01
阻带	6000Hz
阻带波动	0.03

6.20　利用 MATLAB 设计一个 6 阶巴特沃斯低通数字滤波器，抽样频率为 44.1Hz，截止频率为 8kHz。绘制滤波器的相位曲线并与截止频率同样为 8kHz 的 6 阶巴特沃斯低通模拟滤波器比较。关于双线性变换的映射对相位曲线造成的变化，你能得出什么结论？

6.21　以下 MATLAB 语句使用帕克斯-麦克莱伦法生成一个 28 阶 FIR 低通数字滤波器，并满足下列指标。

`F = ([0 .3 .4 1]);`	类别	低通
`M = ([1 1 0 0]);`	抽样频率	1000Hz
`N = 28;`	通带	0～150Hz
`num = firpm(N, F, M);`	通带波动	0.03
`den = 1;`	阻带	200～500Hz
	阻带波动	0.03

利用 MATLAB 构建一个满足相同指标的最小阶数椭圆数字滤波器。分别在同一图中画出两个滤波器的幅度曲线和相位曲线。画出通带和阻带的分解视图并标出阶数。

6.22　已知巴特沃斯模拟滤波器的传输函数如下

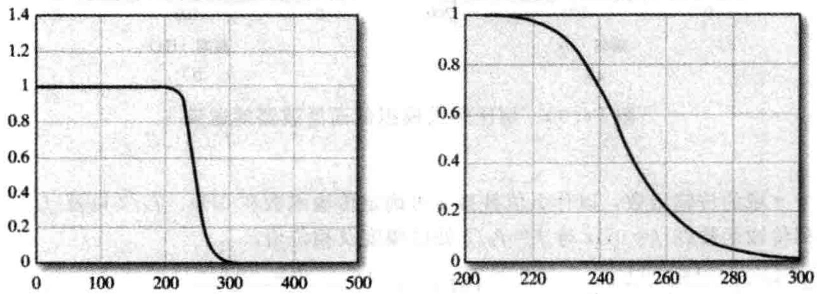
$$H_{\mathrm{B}}(s)=\frac{K}{\prod_{i=1}^{N}(s-p_i)}$$

式中，K 是增益常量，p_i 表示 s 左半平面的极点，可能是复数。

证明当对该传输函数应用双线性变换后，得到的 z 域传输函数在 $z=-1$ 处总有 N 个零点。

6.23　已知一个 6 阶巴特沃斯模拟滤波器，其在 $f=900\mathrm{Hz}$ 处增益为$1/\sqrt{2}$，其在 1900Hz 处增益是多少？

6.24　一个巴特沃斯低通滤波器如图 P6.24 所示。其中给出了滤波器的幅度曲线以及过渡带的放大图。根据图 P6.24 中所示，计算滤波器的阶数。



a)　　b)

图 P6.24　巴特沃斯低通滤波器

6.25　一个 8 阶巴特沃斯模拟滤波器的截止频率为 2500Hz，其在 2575Hz 处的增益是多少？

6.26　巴特沃斯低通模拟滤波器的阶数 N 可由下式得到

$$N=\frac{\ln(K_{\mathrm{S}}/K_{\mathrm{p}})}{2\ln(f_{\mathrm{stop}}/f_{\mathrm{pass}})}$$

式中，$K_{\mathrm{p}}=\dfrac{1}{(1-R_{\mathrm{pass}})^2}-1$，$K_{\mathrm{s}}=(1/R_{\mathrm{stop}}^2)-1$。

(a) 若 $f_{\mathrm{stop}}/f_{\mathrm{pass}}$ 及 R_{stop} 保持不变，假设 $R_{\mathrm{stop}}\ll<1$ 且 $R_{\mathrm{pass}}\ll1$，求 N 与 R_{pass} 的关系。

(b) 若 $f_{\mathrm{stop}}/f_{\mathrm{pass}}$ 及 R_{pass} 保持不变，假设 $R_{\mathrm{stop}}\ll1$ 且 $R_{\mathrm{pass}}\ll1$，求 N 与 R_{stop} 的关系。

(c) 若 R_{pass} 及 R_{stop} 保持不变，假设 $R_{\mathrm{stop}}\ll1$ 且 $R_{\mathrm{pass}}\ll1$，求 N 与 $f_{\mathrm{stop}}/f_{\mathrm{pass}}$ 的关系。

(d) 利用(a)、(b)和(c)得出的关系，判断下列哪些变化将大幅降低滤波器所需的阶数：

(i) 将 R_{pass} 减小 10%

(ii) 将 R_{stop} 减小 10%

(iii) 将 $f_{\mathrm{stop}}/f_{\mathrm{pass}}$ 减小 10%

说明理由。

6.27 求满足下列频率指标的最低阶的巴特沃斯数字低通滤波器，抽样频率为22050Hz：通带从0Hz到1500Hz，波动不高于0.05；阻带从2000Hz到无穷，波动不高于0.01。

6.28 已知某10阶巴特沃斯滤波器在频率492Hz处的幅度响应为0.98，求其截止频率。

6.29 设计一个满足下列指标的切比雪夫数字低通滤波器：

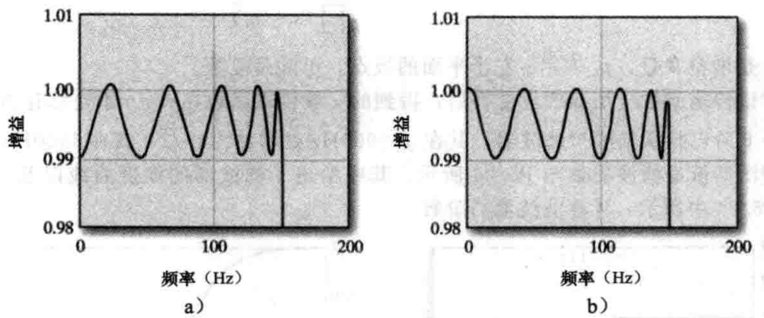
类别	低通
通带	0～2000Hz
通带波动	0.01
阻带	2700Hz～$f_s/2$
阻带波动	0.03
抽样频率	11025Hz

利用MATLAB中的chirp函数生成一个扫频余弦波(swept cosine wave)如下：

```
t=0:T:2; % 2 secs @ fs sample rate
u=chirp(t,0,2,fs/2, 'linear');
```

利用MATLAB中的filter函数将该扫频余弦波应用于切比雪夫滤波器。绘制filter函数的时域输出。该时域输出与切比雪夫滤波器的幅频特性曲线存在什么关系？

6.30 图P6.30所示为切比雪夫模拟低通滤波器的通带，求解滤波器的阶数并加以说明。



图P6.30 切比雪夫模拟低通滤波器的通带

6.5节

6.31 给出一个z域的传输函数，以什么值替换z可确定传输函数在0Hz、$f_s/2$以及$f_s/4$处的增益？

6.32 求出下列传输函数在$f=0$Hz与$f=f_s/2$处的幅度及相位值。

(a) $H(z)=\dfrac{5(z+1)}{z^2+1.4z+0.2}$　　(b) $H(z)=\dfrac{5(z-1)}{z^2+1.4z+0.2}$

(c) $H(z)=\dfrac{z^3}{(z-1)^3}$

6.33 求K的值，使得0Hz处的$|H(z)|=1$。

$$H(z)=\frac{K(z+1)^3}{z^3-1.585z^2+1.116z-2.890}$$

6.34 (a) 试证明任何z域传输函数，其为实系数且在$z=+1$处没有极点和零点，则其在0Hz处必有0°或180°的相移。

(b) 试证明任何稳定系统的z域传输函数，其为实系数，且零点在单位圆内或单位圆上，并在$z=+1$处有N个零点，则其在0Hz处必有±0°的相移。

6.35 假设有一个稳定的数字滤波器，若令传输函数中的$z=1$，则函数值为−15。

(a) 在0Hz处的相移是多少？

(b) 脉冲响应的终值为多少？

(c) 若阶跃信号的幅值为0.5，则其阶跃响应的终值为多少？

6.36 信号$x(t)=2\sin(2\pi 5000t)$经过传输函数为$H(z)=\dfrac{z+1}{z^2-\sqrt{3}/2z+1/2}$的系统。若抽样频率为20000Hz，求输出信号的幅度及相位。

6.37　已知传输函数如下

$$H(z)=\frac{z-1}{z^2+1.2z+0.52}$$

回答下列问题：

(a) 该滤波器是低通、高通、带通还是带阻？　(b) 该滤波器的阶数是多少？

(c) 系统是否稳定？　(d) 求该滤波器对应的差分方程。

(e) 该滤波器阶跃响应的终值是多少？　(f) 该滤波器脉冲响应的前两项是什么？

6.38　已知传输函数如下

$$H(z)=\frac{z+1}{z^2-1.2z+0.52}$$

回答下列问题：

(a) 该滤波器是低通、高通、带通还是带阻？

(b) 该滤波器的阶数是多少？

(c) 系统是否稳定？

(d) 求该滤波器对应的差分方程。

(e) 该滤波器阶跃响应的终值是多少？

(f) 该滤波器脉冲响应的前两项是什么？

6.39　假设有一个信号，其幅度变化范围位于＋1～－1V，该信号被频率为 60Hz 幅度为±10V 的噪声干扰。若要将噪声降到信号幅度的±2%以内，滤波器的通带和阻带的衰减应为多少分贝？

6.40　一个与你有浪漫关系的人发给你一个 .wav 格式的文件，这是来自南太平洋诸岛的音乐。遗憾的是，它被 60Hz 的电力线噪声所干扰。若信号与噪声的幅度范围都是位于＋1V～1V 之间，而你要将这 60Hz 的电力线噪声降到信号幅度的 2%以内，阻带衰减应为多少分贝才能保持这段浪漫史？

6.41　对图 P6.41 中每个 z 平面的零极点图，使用几何分析法绘制各自的幅度和相位的近似曲线。

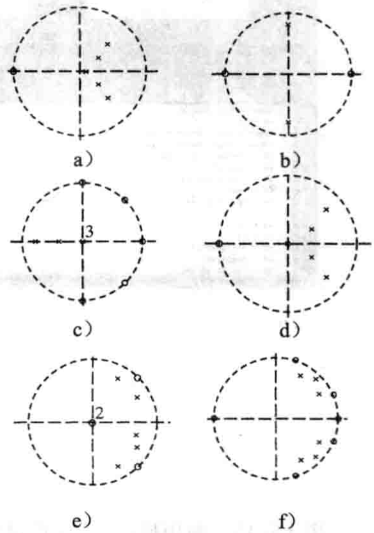


图 P6.41　确定每个极/零点图的近似幅度和相位曲线

6.42　已知一个极/零点图，其在 $z=-1$ 处有一个零点，在 $z=\pm j$ 处有复零点，且在原点有三个极点。写出两个拥有这样零极点图的其他传输函数。

6.43　构建一个系统，其在 $0.6\pm j0.5$ 处有一个复零点且在原点有两个极点。写出三个与该系统具有同样幅度曲线的传输函数。

6.44　图 P6.44 所示为两个系统的频率响应曲线。针对每个曲线回答下列问题：

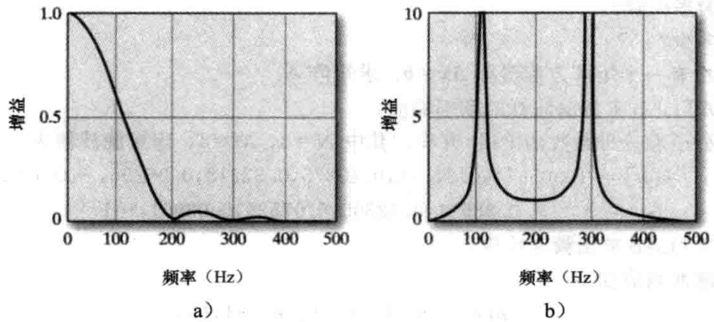


图 P6.44　两个系统的频率响应曲线

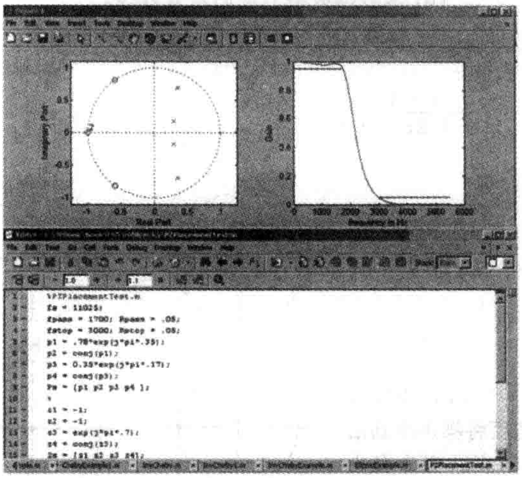
(a) 由图 P6.44 可推断出哪些零点？　(b) 由图 P6.44 可推断出哪些极点？

(c) 系统是否稳定？　(d) 系统是否因果？

(e) 它是 FIR 滤波器还是 IIR 滤波器？

6.6 节

6.45 MATLAB 中的 poly 函数可将极点或零点向量转变为可在频率空间表示的多项式。创建 m 文件，其输出图像如图 P6.45 所示。若调整 m 文件中的部分值，即可改变极点和零点的位置，且可以从频率空间中看到改变的效果。利用此方法设计一个满足以下指标的低通滤波器：抽样频率 $f_s=11025$Hz，通带边界为 2000Hz，阻带边界为 2800Hz，通带及阻带波动为 0.05。



```
fs = 11025;
fps = 1700; Rps = .05;
fst = 3000; Rst = .05;
p1 = .78*exp(j*pi*.35);
p2 = conj(p1);
p3 = 0.35*exp(j*pi*.17);
p4 = conj(p3);
Ps = [p1 p2 p3 p4 ];
%
z1 = -1;
z2 = -1;
z3 = exp(j*pi*.7);
z4 = conj(z3);
Zs = [z1 z2 z3 z4];
den = poly(Ps);
num = poly(Zs);
[H f] = freqz(num,den,1024,fs);
figure(1);clf
subplot(1, 2, 1);
zplane(num, den);
subplot(1, 2, 2);
plot(f, abs(H)/max(abs(H)));
xlabel('frequency in Hz');
ylabel('Gain');
line([0 fps],[1-Rps 1-Rps]);
line([fst fs/2],[Rst Rst]);
```

图 P6.45 利用极/零点位置设计滤波器的 MATLAB 界面。编辑器为 m 文件，p_s 和 z_s 分别是极点和零点组成的向量。改变极点和零点的位置后运行 m 文件即可得到极/零点分布及其频率响应

6.46 某理想滤波器的脉冲响应为

或 $|h[n]|=(1/2)^n$

利用 Pade 法设计一个数字滤波器，$f_s=11025$Hz。选取 $M=N=4$。回答下列问题：

(a) 需要多少个方程？

(b) 滤波器为多少阶？

(c) b_0 值是多少？

(d) Pade 法中有一个矩阵方程形如 $\boldsymbol{Ax}=\boldsymbol{b}$。求矩阵 $\boldsymbol{A}$。

(e) 利用 MATLAB 求解滤波器的频率响应。

6.47 写出滤波器分子和分母系数的 Pade 方程，其中 $N=3$，$M=3$，理想滤波器为

$$h[n]=\{0.05874,0.24544,0.40875,0.32218,0.06298,-0.10723,\\-0.08514,0.02305,0.07536,0.03961,\cdots\}$$

利用 MATLAB 求出频率响应。

6.48 理想滤波器脉冲响应如下

$$h(nT)=\{h0,h1,h2,h3,h4,h5,\cdots\}$$

若 $M=1$，$N=2$，利用 Pade 法写出 IIR 滤波器系数的矩阵方程。

6.49 利用 Pade 法生成了下列矩阵的方程组。回答下列问题：

(a) M 与 N 分别是多少？

(b) 生成这组方程的理想脉冲响应前 6 项是什么？

(c) 利用这组方程构建的滤波器的脉冲响应前 6 项是什么？

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & -1 & 0 & 0 \\ 0 & 0 & 1 & -0.7 & -1 & 0 \\ 0 & 0 & 0 & -0.1 & -0.7 & -1 \\ 0 & 0 & 0 & 0.2 & -0.1 & -0.7 \\ 0 & 0 & 0 & 0.1 & 0.2 & -0.1 \end{bmatrix} \cdot \begin{bmatrix} b_0 \\ b_1 \\ b_2 \\ a_1 \\ a_2 \\ a_3 \end{bmatrix} = \begin{bmatrix} -1 \\ -0.7 \\ -0.1 \\ 0.2 \\ 0.1 \\ 0.05 \end{bmatrix}$$

6.7 节

6.50　一个交流电动机使用电压为 120V、频率为 60Hz 的交流电源供电。电枢电阻与电枢温度成比例。为了确定电动机旋转(动态)时的温度，给电枢输入一个小的直流电。希望通过去除电枢电压的交流分量来恢复直流电压。直流电压约为 100mV，所要求的测量误差不能高于±5%。确定适当的滤波器指标。

6.51　低通滤波器的截止频率常设定在幅度曲线的 3dB 处。证明 3dB 处对应半功率点。

6.52　假若增益在线性刻度下按照 10 的倍数改变，则在 dB 刻度下相应地按照多少变化?

6.53　为了说明梳状滤波器的应用，利用 MATLAB 生成一个带噪声的方波并将其通过一个梳状滤波器，该梳状滤波器峰值增益出现在方波的基波和奇数谐波的频率位置。以下 MATLAB 代码可用于生成方波并叠加噪声。

```
%NoisySquare.m
fs = 44100;T = 1/fs; % Sample frequency is 44,100 Hz
f0 = fs/20;T0 = 1/f0; % Square wave at 2,205 Hz
t = 0:T:40*T0;
SqWv = square(2*pi*f0*t);

figure(1);clf;
subplot(2, 1, 1);
plot(t,SqWv);
axis([0 10*T0 -1.5 1.5]);
% Add Noise to the wave
Noise = .75*(rand(1, length(SqWv))-.75);
NoisWv = SqWv + Noise;
subplot(2, 1, 2);
plot(t,NoisWv);
axis([0 10*T0 -1.5 1.5]);
```

因为本题中的方波基频是抽样频率的 1/20，所以可用 10 阶梳状滤波器。采用 $r=0.99$ 的值得到梳状滤波器的传输函数如下

$$H_{\text{comb}}(z) = K\frac{z^{10}-1}{z^{10}+0.99^{10}}$$

利用 MATLAB 中的 filter 函数对带噪声的方波信号进行滤波。画出带噪方波信号及其幅度谱，滤波器的频率响应以及滤除噪声后的方波信号的幅度谱。解释为什么此过程没有完全滤除方波信号的噪声。

6.54　下列 MATLAB 代码生成一个频率为 80Hz 并含有白噪声的方波信号的样点序列，带宽约为 1000Hz。设计一个梳状滤波器恢复该方波 500Hz 内的基波及其奇数谐波。利用 MATLAB 中的 filter 函数仿真方波通过梳状滤波器的结果，并画出含噪声的原方波及降噪后的方波。

```
t = 0:.001:2.5;
% 30 Hz, 50% duty cycle 0 to 1.
y = (square(2*pi*80*t,50) + 1)/2;
% White noise
y = y + (rand(1,length(y)) - .5);
% Band limit with 4th order Butterworth filter
[num den] = butter(4,2*pi*500,'s');
sys = tf(num,den);
% lsim simulates passing y through Butterworth filter
yf = lsim(sys,y,t);
% yf is the band limited square wave vector
```

利用与习题 6.53 类似的梳状滤波器，r 值为 0.995。

6.55 对于一个截止频率为 12000Hz、抽样频率为 44100Hz 的 3 阶巴特沃斯数字滤波器，其相位曲线在通带非线性。试证明通过级联一个全通滤波器可对其进行修正，全通滤波器的传输函数为

$$H_A(z)=\frac{-0.4(z-(1/r))}{z-r}$$

式中，$r=0.4$。

利用 MATLAB 画出原巴特沃斯数字滤波器及其级联后滤波器的幅度和相位曲线。

6.56 一个 2 阶全通数字滤波器的传输函数为

$$H_A(z)=K\frac{z-2.5}{z-0.4}\cdot\frac{z-2.5}{z-0.4}$$

(a) 这个滤波器对于近似的频带中心造成的相移是多少?

(b) K 值为多少时使整个频带的增益为 1?

6.57 已知一个单级全通数字滤波器的传输函数为

$$H(z)_{\mathrm{AllPass}}=\frac{0.8z+1}{z+0.8}$$

回答下列问题：

(a) 该滤波器在整个频带的增益是多少?

(b) 若两个这样的单级全通滤波器级联为一个二级全通滤波器，则其对近似的频带中心造成的最大相移是多少?

6.58 利用几何方法说明，谐振器与陷波滤波器关于带宽有几乎相同的计算方程。

6.59 求出下面陷波滤波器的带宽。选取 $f_s=10000\text{Hz}$。

$$H(z)=\frac{z^2-1.4142z+1}{z^2-1.3435z+0.9025}$$

6.60 一个陷波滤波器在 400Hz 处的衰减为 92dB，频带宽度为 20Hz。若在 400Hz 处有幅度为 1V 的噪声，经过该滤波器后噪声的幅度是多少?

6.61 假如我们需要设计一个陷波滤波器来去除 60Hz 的电力线噪声，抽样频率为 1000Hz，带宽为 4Hz，传输函数如下：

$$H(z)=K\frac{z^2-1.8596z+1}{z^2-1.8362z+0.97502}$$

(a) 若电力线的频率实际为 59.5Hz 而不是 60Hz，估算此频率点的分贝衰减。

(b) 重新设计陷波滤波器，使得当陷波中心为 60Hz 时，59.5Hz 处至少有 20dB 的衰减。

6.62 某陷波滤波器的传输函数为

$$H(z)=\frac{z^2-1.732050808z+1}{z^2-1.69740979z+0.9604}$$

若抽样频率 $f_s=11025\text{Hz}$，该陷波滤波器的中心频率和带宽是多少?

6.63 已知某谐振器在 $z=1$ 及 $z=-1$ 处有零点，$z=0.49018+\mathrm{j}0.84902$ 处有极点。若抽样频率为 8000Hz，该谐振器的中心频率和带宽是多少?

6.64 试求数字谐振器的 z 域传输函数，已知该谐振器有一对复极点和一对复零点，抽样频率为 300Hz，谐振频率在 60Hz 处且带宽为 2Hz。

6.65 试求一个简单谐振器的极点和零点位置，已知中心频率为 400Hz，带宽为 20Hz，抽样频率为 3000Hz。结果以矩形格式排列。

6.66 若低通和高通滤波器级联，且有相同的截止频率，结果将为带通滤波器，其增益在截止频率处为 0.5。

(a) 根据以上所述，采用一个 12 阶低通和一个 12 阶高通的巴特沃斯滤波器设计一个带通滤波器，其中心频率为 2000Hz。归一化带通滤波器使得最大增益为 1，并利用 MATLAB 画出幅度曲线。抽样频率为 11025Hz。

(b) 若高通滤波器的截止频率上升而低通滤波器的截止频率下降，(a)所设计的带通滤波器形状将会如何变化?

(c) 若高通滤波器的截止频率下降而低通滤波器的截止频率上升，(a)所设计的带通滤波器形状将会如何变化? 根据低通和高通滤波器的截止频率描述所得带通滤波器的带宽，说明带通滤波器的阻带波动与低通和高通滤波器的阻带波动的关系。

6.67　若低通和高通滤波器并联，且低通滤波器的截止频率低于高通滤波器的截止频率，则结果为带阻滤波器。

(a) 根据以上所述，采用一个 12 阶低通和一个 12 阶高通的巴特沃斯滤波器设计一个带阻滤波器，其中心频率为 2000Hz。设置低通滤波器的截止频率为 1000Hz 而高通滤波器的截止频率为 3000Hz。利用 MATLAB 画出所得带阻滤波器的幅度曲线。抽样频率为 11025Hz。

(b) 若高通滤波器的截止频率低于低通滤波器的截止频率，(a)所设计的带阻滤波器形状将会如何变化？

6.68　现有两个数字滤波器 $H_1(z)=\dfrac{N_1(z)}{D_1(z)}$，$H_2(z)=\dfrac{N_2(z)}{D_2(z)}$ 相并联，相比原来的两个滤波器，该并联滤波器的极点将会出现在哪里？

6.69　一个带有 8 位 A/D 转换器的系统用以实现数字带阻滤波器。该带阻滤波器的阻带增益比通带低 90dB，通带波动为 0.05。这样安排有什么问题？

6.70　若 $h[n]=\{1\ 2\ 2\ 1\}$，$g[n]=\{1\ 3\ 4\ 3\ 1\}$，试证明卷积 $h[n]*g[n]=p[n]$可看作两个多项式的乘积，其中，$h[n]=z^3+2z^2+2z+1$，而 $g[n]=z^4+3z^3+4z^2+3z+1$。

6.71　某 z 域传输函数如下，求 K_1 和 K_2 的值，使得系统在 1000Hz 时为振荡器，假如抽样频率为 11025Hz。利用 MATLAB 求解脉冲响应，并确认系统在 1000Hz 时振荡。

$$H(z)=\frac{z^2-1}{z^2+K_1z+K_2}$$

6.72　设计一个 11 阶椭圆滤波器，通带为 0～150Hz，通带和阻带波动为 0.001，抽样频率为1000Hz。以该滤波器的脉冲响应(前 30 项)作为 Prony 法的理想滤波器。利用 MATLAB 以 Prony 法构建数字滤波器，其分子与分母有相同的阶数，且阶数从 8 变化到 11。在同一图中绘制不同阶数所对应滤波器的响应及其理想滤波器的响应。

6.73　已知 RLC 电路的脉冲响应为

$$h(t)=\mathrm{e}^{-t/\tau}[2\cos(\omega_\mathrm{d}t)+3\sin(\omega_\mathrm{d}t)]$$

其中，$\omega_\mathrm{d}=2000\pi$，$\tau=0.001$。

(a) 选择抽样频率为 10000Hz，并在时域画出脉冲响应曲线。

(b) 利用 MATLAB 的 deconv 函数求其逆滤波器的脉冲响应。并在时域画出逆滤波器的脉冲响应曲线。

(c) 利用 Prony 法设计 IIR 逆滤波器，其中 $M=N=2$。

6.74　数字滤波器的实现方程由以下表示式给出

```
y = x - x2;
y = y - 1.84*y1 - .9678*y2;
x2 = x1;x1 = x;
y2 = y1;y1 = y;
```

(a) 求解此滤波器的传输函数。

(b) 指出该滤波器是全通、低通、高通、带通还是带阻？

6.75　以下 MATLAB代码生成长度为 1s 的 sinc 脉冲串。该信号被线性调频(chirp)噪声干扰。

```
sinc1 = @(t1) sinc(100*t1);
fs = 10000;T = 1/fs;
t = 0:T:1;
d = [(.1:.1:.8); exp(-.5:-.5:-4)]';
ptrain = 2*pulstran(t, d, sinc1);
y = ptrain;
yn = ptrain + chirp(t, 100, 1, 4500);
```

利用满足下列指标的滤波器可有效滤除噪声：通带和阻带波动＝0.01；通带边界 100Hz；阻带边界 150Hz，抽样频率 10kHz。设计满足指标且阶数最小的滤波器。利用 MATLAB 中的 filter 函数滤除噪声以恢复信号，并在同一幅图中画出原始信号及恢复后的信号。

第7章

抽样速率转换

在进行模拟-数字转换时，常在DSP系统的前端使用过抽样技术，从而降低抗混叠滤波器的要求，而在DSP系统的末端，通常也会使用过抽样技术(在4.2节中的抗混叠滤波中已有介绍)以降低抗镜像滤波器的要求。在信号只包含较低频率信息的情形下，应降低其抽样速率，该操作称为抽取，也称为降抽样。

对信号进行降抽样时，注意不要产生混叠，当新的抽样频率小于信号最高频率两倍时，会产生频谱混叠现象。而对信号进行升抽样时，新的信号频谱将包含原信号的镜像频谱。

在集成系统中，经常会出现以某一速率抽样的信号输入到另一不同抽样速率的模块的情况，因而降抽样和升抽样具有重要的作用。本章将详细介绍抽样速率改变的频域特征及其实现过程，同时探讨已抽样的输入信号在其抽样速率改变时将会发生怎样的变化。

7.1 整数抽取

抽取一词来源于拉丁文，其原意为“每十个中去除一个”，正如很多英文单词一样，由于用来表示其他意思太久以至于其他意思已经取代其原意。在数字信号处理中，抽取的意思是指以某一数值降低抽样速率，该值通常为整数。如果降低抽样速率，奈奎斯特频率降低相同的倍数，那么之前抽样时信号频谱没有出现混叠，在降低抽样速率后信号频谱中可能出现混叠。因而，需要在信号降抽样前加入低通抗混叠滤波器，称其为抽取器。

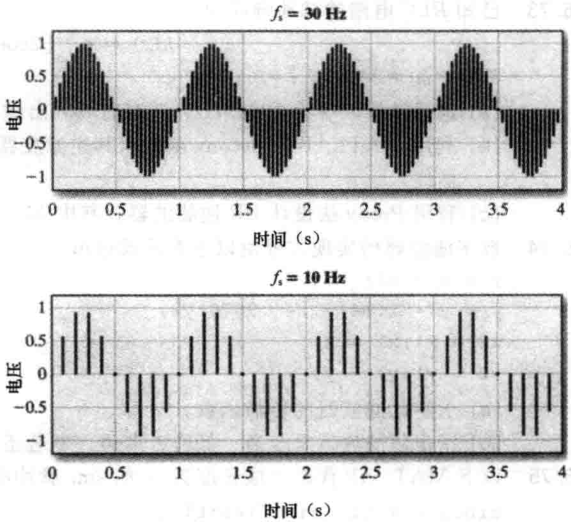


图7.1 每三个点去除两个抽样点实现3倍抽取

整数倍抽取可通过去除信号中部分抽样点来实现。例如：为实现3倍抽取，则每三个点中去除两个抽样点，如图7.1所示。

直觉上，当进行D倍抽取时，采用D个抽样点的平均值作为新的抽样值比直接删除前$D-1$个抽样点要好。这个处理过程等价于将信号先通过一个长度为D的滑动平均滤波器，然后每D个抽样点删除$D-1$个抽样点。正如第5章所述，尽管滑动平均滤波器计算效率很高，但其并不是一个很好的低通滤波器。

软件上，可以使用模除运算实现抽取，使用计数器跟踪需要保存哪个抽样点，需要删除哪个抽样点。降抽样的伪代码实现如下：

```
unsigned integer count = 0;
loop every T seconds      //T is the original signal period
  if(count mod D is zero) //D is the down-sample factor
    Get this sample → x
    output x
  count = count + 1
bottom of loop
```

抽取器前端的抗混叠滤波器用来滤除可能产生混叠效应的频率。这里的抗混叠滤波器是一个数字滤波器——不是在抽样之前滤除频率大于 $f_s/2$ 的那个模拟滤波器。如果原信号的抽样频率为 f_s，进行 D 倍降抽样，则抗混叠滤波器需滤除信号中大于 $f_s/(2D)$ 的频率。理想的抗混叠滤波器具有以下频率响应特性：

$$H_{AA}(f)=\begin{cases}1, & 0\leqslant f\leqslant f_s/(2D)\\ 0, & \text{其他}\end{cases}$$

式中，D 为抽取倍数。

抽取不是一个时不变过程。时不变系统要求输入信号的时移在其输出端产生相同的时移。输入信号 $x[n]$，产生的输出为 $y[n]$，若系统为时不变系统，则信号 $x[n-k]$ 产生的输出为 $y[n-k]$。对于抽取器，$y[n]=x[nD]$，但是 $y[n-k]=x[(n-k)D]$ 并不等于 $x[nD-k]$，因此抽取不是时不变系统，但其是线性系统。

例 7-1 chirp 信号的创建和抽样

使用 MATLAB 创建一个 chirp 信号，其频率范围为 10～2000Hz，抽样率为 11025Hz，不通过滤波分别进行 2 倍和 4 倍降抽样，观察混叠效应。

解：原信号使用如下 MATLAB 代码产生：

```
N = 1000;
fs = 11025;T = 1/fs;
t = 0:T:(N-1)*T;
y = chirp(t, 10, (N-1)*T, 2000);
subplot(1, 2, 1);
stem(t, y, 'MarkerSize', 1);
title('Chirp signal');
subplot(1, 2, 2);
yDFT = fft(y);
f = 0:fs/N:fs-fs/N;
yDFTNorm = abs(yDFT)/max(abs(yDFT));
plot(f, yDFTNorm);
```

在 MATLAB 中，可以使用 downsample 函数来不通过滤波降低抽样频率：

```
yDn = downsample(y, 2); %Down-sample by a factor of 2.
```

结果如图 7.2 所示。◀

7.1.1　降抽样后信号的频谱

$x[nT]$ 为连续时间信号 $x(t)$ 的抽样后信号，抽样频率为 ω_s，则 $x[nT]$ 的频谱为：

$$X(e^{j\omega T})=\frac{1}{T}\sum_{n=-\infty}^{\infty}X_{cont}[j(\omega-n\omega_s)]$$

式中，$X_{cont}(j\omega)$ 是 $x(t)$ 的频谱[1,2]，由此可见，$x[nT]$ 的频谱是 $X_{cont}(j\omega)$ 的周期化，周期为 ω_s。

如果对信号 $x[nT]$ 进行 D 倍降抽样，D 为整数，新的抽样频率 $\omega_{sD}=\omega_s/D$，降抽样后信号为 $x_d[nT_D]$，其中 $T_D=T\times D$ 为新的抽样周期。

$x_d[nT_D]$ 的频谱为：

$$X_d(e^{j\omega T_D})=\frac{1}{TD}\sum_{n=-\infty}^{\infty}X_{cont}[j(\omega-n\omega_{sD})]$$

从以上公式可见，降抽样后信号 $x_d[nT_D]$ 的频谱仍是 $X_{cont}(j\omega)$ 的周期化，但其周期为 ω_{sD}。

图 7.3 所示为一个带限信号的 3 倍降抽样。

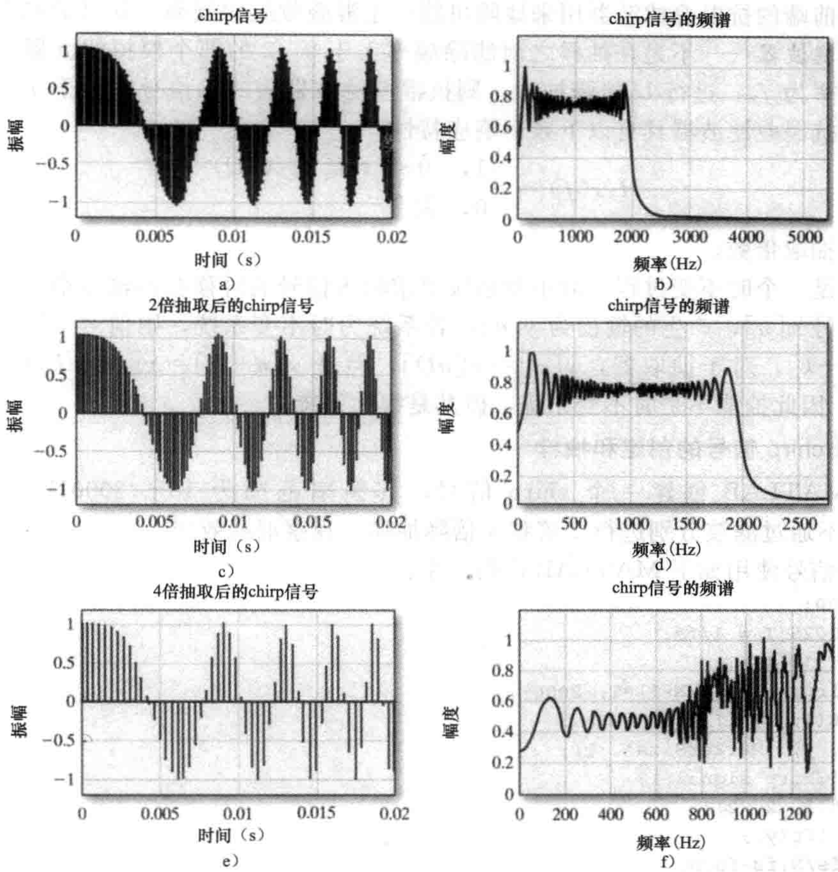


图 7.2　图 a 为原始 chirp 信号，其抽样率为 11025Hz，图 b 为原始信号的频谱，图 c 和 d 为 2 倍抽取后信号的时域和频域表示，由于信号大部分频率低于 $f_s/2$，仅有少量混叠。图 e 和 f 为 4 倍抽取后信号的时域和频域表示，频谱存在明显的混叠

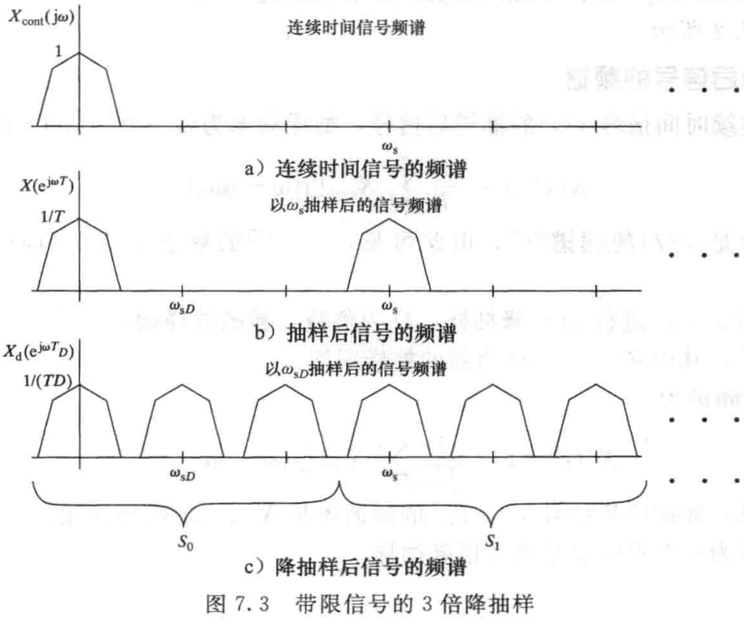


图 7.3　带限信号的 3 倍降抽样

从图 7.3 中可以看出，可以将降抽样后的信号频谱用集合来表示，每一个集合包含 D 个与原信号频谱相同的波形，集合 S_0 可写为：

$$\frac{1}{TD}\sum_{n=0}^{D-1} X_{\text{cont}}[\mathrm{j}(\omega - n\omega_{\text{s}D})]$$

从上式可以看到，ω 平移了 $n\omega_{\text{s}D}$。将所有集合平移 $k\omega_{\text{s}}$ 后相加：

$$X_{\text{d}}(\mathrm{e}^{\mathrm{j}\omega T_D}) = \sum_{k=-\infty}^{\infty} \frac{1}{TD}\sum_{n=0}^{D-1} X_{\text{cont}}[\mathrm{j}(\omega - k\omega_{\text{s}} - n\omega_{\text{s}D})]$$

交换求和的顺序，得到：

$$X_{\text{d}}(\mathrm{e}^{\mathrm{j}\omega T_D}) = \frac{1}{D}\sum_{n=0}^{D-1}\left(\frac{1}{T}\sum_{k=-\infty}^{\infty} X_{\text{cont}}[\mathrm{j}(\omega - k\omega_{\text{s}} - n\omega_{\text{s}D})]\right)$$

由于

$$X(\mathrm{e}^{\mathrm{j}\omega T}) = \frac{1}{T}\sum_{n=-\infty}^{\infty} X_{\text{cont}}[\mathrm{j}(\omega - n\omega_{\text{s}})]$$

得到

$$X_{\text{d}}(\mathrm{e}^{\mathrm{j}\omega T_D}) = \frac{1}{D}\sum_{n=0}^{D-1} X[\mathrm{e}^{\mathrm{j}(\omega - k\omega_{\text{s}} - n\omega_{\text{s}D})}]$$

该公式给出了降抽样信号频谱相对于原信号频谱的数学表达式。

7.1.2　级联抽取

在许多实际应用中，通常需要进行较大倍数的抽取，如大于 10。在该情况下，相比单级抽取，以多级级联的方式实现抽取更可行，计算效率更高。N 级抽取器级联的实现方案如图 7.4 所示。

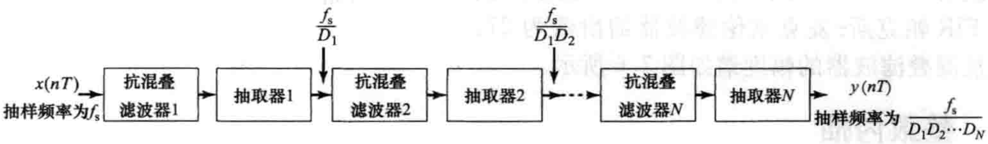


图 7.4　N 级级联抽取器结构图，每一级的整数抽取均有单独的抗混叠滤波器

例 7-2　二级级联抽取器的实现

信号的通带为 0～100Hz，过渡带为 100～150Hz，抽样频率为 20000Hz，希望将抽样频率降低到 1000Hz，也即以 $D=20$ 抽取，为该抽取器设计抗混叠滤波器，要求其通带波动 $r_{\text{p}}=0.01$，阻带波动 $r_{\text{s}}=0.001$。

解： 采用 MTALAB 计算单级抽取时滤波器的阶数，当采用帕克斯-麦克莱伦滤波器时，计算所得阶数为 339。

```
fs = 20000;
fpass = 100;
fstop = 250;
rpass = 0.01;
rstop = 0.001;
F = ([fpass fstop]);
M = ([1 0]);
Dev = [rpass rstop];
[N F A W] = firpmord(F, M, Dev, fs);
disp(N);
```

使用二级级联抽取解决该问题，可以考虑的抽取器组合有：2-10，10-2，4-5 或 5-4。图 7.5给出了 5-4 组合的二级级联抽取器结构图。

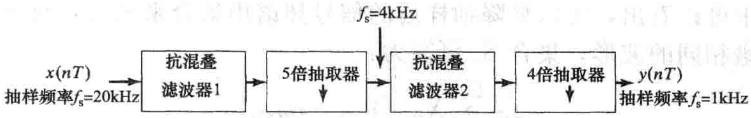


图 7.5　二级级联抽取器，第一个 5 倍抽取器将抽样频率由 20kHz 降为 4kHz，第二个 4 倍抽取器将抽样频率由 4kHz 降为 1kHz

第一个抗混叠滤波器的输入信号抽样频率为 20kHz，5 倍抽取后，输出信号的抽样频率降为 4kHz，该滤波器需滤除可能产生混叠的频率，即大于 2kHz 的频率。因此，对于第一个抗混叠滤波器，通带 $f_{pass}=0\sim100\text{Hz}$，阻带 $f_{stop}=2\text{kHz}\sim f_s/2$。

通带波动增益为两个抗混叠滤波器波动的乘积，因而需要降低每一个滤波器的通带波动以保证整个系统的波动满足设计要求。级联后抗混叠滤波器的通带增益为$(1-r_{p1})^2=1-r_p$，即

$$1-2r_{p1}+r_{p1}^2=1-r_p$$

由于波动很小，忽略平方项得到：

$$r_{p1}=r_p/2$$

对于此例，阻带波动无须改变。使用上面的参数和例 7-2 中的 MATLAB 代码，帕克斯-麦克莱伦滤波器所需的阶数为 28。

第二个抽取器的输入信号抽样频率为 4kHz，输出信号的抽样频率为 1kHz，其抗混叠滤波器的通带为 0～100Hz，阻带为 500～$f_s/2$，通带波动=0.005，阻带波动=0.001，此参数设计的 FIR 帕克斯-麦克莱伦滤波器的阶数为 27。两个抗混叠滤波器的幅度谱如图 7.6 所示。

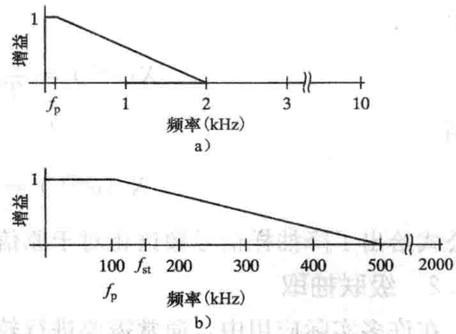


图 7.6　抗混叠滤波器的幅度谱，对于图 a，通带仍保持在 100Hz，阻带从 $f_{st1}=2\text{kHz}$ 开始

7.2　整数内插

内插与抽取相反，其以整数倍*增加*信号的抽样频率。内插通常为两点之间画一条直线，沿着直线计算中间点的值。但在数字信号处理中，内插滤波器是在抽样点之间插入零以增加信号的抽样频率，然后将该结果通过低通滤波器以消除产生的镜像效应。下面将探讨在抽样点之间插入零后，信号频域的变化。

通过内插零，将以 10Hz 抽样后的 1Hz 的正弦信号升抽样到 20Hz，观察抽样后正弦信号的时域和频域，如图 7.7 所示。在频域中，可以看到在 $f_s/2\sim f_s$ 之间，出现了镜像频率。图 7.7a 是 1Hz 的原始正弦信号，图 7.7b 为以 10Hz 抽样后信号的频谱，1Hz 的正弦信号在 9Hz 处出现了镜像。图 7.7c 中，在抽样点之间插入零，抽样频率变为原来的 2 倍，图 7.7d 为升抽样后信号的频率响应，所示为原信号和其镜像(9Hz)的叠加。

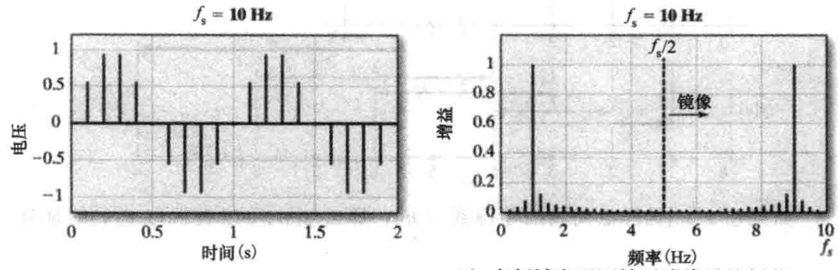
图 7.7c 所示为抽样点间内插零后产生的效果，图 7.7d 所示为得到的频率响应，其并不直观。而图 7.8 显示，如果将 1Hz 和 9Hz 的正弦信号相加，并以 20Hz 的抽样频率进行抽样，将得到与图 7.7c 相同的结果。

如果信号 $x[n]$如图 7.9 所示，在抽样点之间内插 0 后产生新的信号 $y[n]$，以数学形式表达为：

$$y[0]=x[0]$$
$$y[1]=0$$
$$y[2]=x[1]$$
$$y[3]=0$$

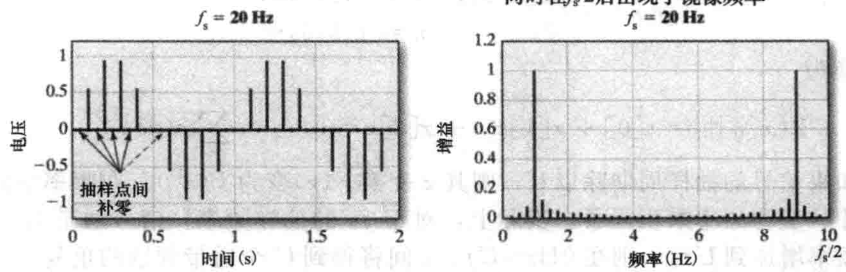
$$y[4] = x[2]$$

…



a）抽样间隔为0.1s的1Hz的正弦信号

b）在频域出现原始正弦信号的频率，同时在f_s/2后出现了镜像频率



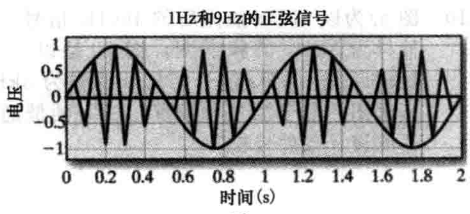
c）在a的抽样点之间内插零后的正弦信号，抽样周期降为0.05s

d）内插零后的信号抽样频率增加为原来的2倍，且在9Hz的地方出现原始信号的镜像

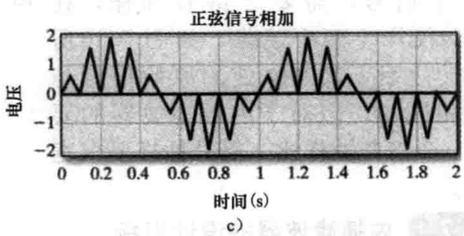
图 7.7　内插零后抽样正弦信号的时域和频域变化

```
fsig = 1;                          %signal
fs = 20;T = 1/fs;
n = 0:4*fsig/T;
x = sin(2*pi*fsig*n*T);
x1 = sin(2*pi*fsig*9*n*T);
subplot(2, 1, 1);
  t = 0:.05:4;
  plot(t, x);
  hold on;
  plot(t, x1, 'r');
  xlabel('Time in seconds');
  ylabel('voltage');
  title('Sinusoids at 1 Hz and 9 Hz');
subplot(2, 1, 2);
  plot(t, x+x1);
  xlabel('Time in seconds');
  ylabel('voltage');
  title('Sinusoids added');
```

a）



b）



c）

图 7.8　图 b 所示为 1Hz 和 9Hz 的正弦信号，图 c 为二者之和，如果以 0.05s 对图 c 抽样，将得到与以 0.1s 抽样的 1Hz 正弦信号，且在抽样点间内插零相同的结果

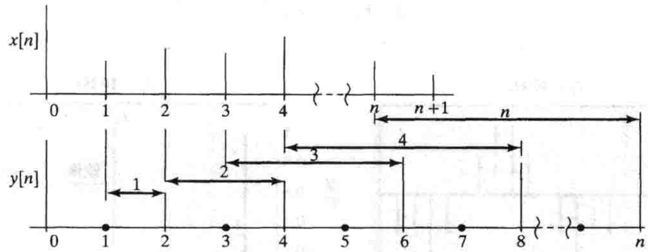


图 7.9 如果在 $x[n]$抽样点间内插零得到 $y[n]$，那么 $y[n]$可以表示为 $x[n]$的延时

一般形式为

$$y[n]=\begin{cases}x[n/2]; & n=0,2,4,6,\cdots\\0; & n=1,3,5,\cdots\end{cases}$$

$x[n]$的 z 变换为

$$Z\{x[n]\}=x[0]+x[1]z^{-1}+x[2]z^{-2}+\cdots=\sum_{n=0}^{\infty}x(n)z^{-n}$$

一般来说，如果信号的抽样周期除以 U，则其 z 变换 $F(z)$变为 $F(z^U)$。在频率空间中，用 $e^{j\Omega}$代替 z，则 U 变为频率乘积因子。实际上，对于 f_s 的抽样频率，信号通带为 $0\sim f_s/2$，如果将抽样频率增加到 Uf_s，则在 0Hz～$Uf_s/2$ 间将得到 U 个通带频谱的重复。

内插包含在信号的抽样点间插入零，并将结果通过低通滤波器以去除镜像效应。用于去除镜像效应的滤波器称为内插滤波器，其与抗混叠滤波器是类似的理想的内插滤波器具有以下特性：

$$H_{\text{Ideal}}(f)=\begin{cases}U, & 0\leqslant f<f_s/(2U)\\0, & f_s/(2U)\leqslant f\leqslant f_s/2\end{cases}$$

式中，f_s 为内插前的信号抽样频率，U 为升抽样因子。理想内插器的增益为 U，用于补偿内插零后信号平均值的降低。

例 7-3 升抽样

假设有两个相同的信号，其为 100Hz 附近带宽很窄的信号，第一个信号的抽样频率为 1kHz，第二个信号的抽样频率为 1.5kHz，如果将两个信号的抽样频率提高到 3kHz，内插后的两个信号的频谱有什么不同？

解： 对于第一个信号，需要 3 倍升抽样，在 0～3kHz 将产生 3 个镜像。对于第二个信号，需要 2 倍升抽样，在 0～3kHz仅产生两个镜像，如图 7.10所示。滤波之后，100Hz 处的原始信号与之前是相同的，但对于内插滤波器的要求而言，2 倍升抽样比 3 倍升抽样要求宽松一些。

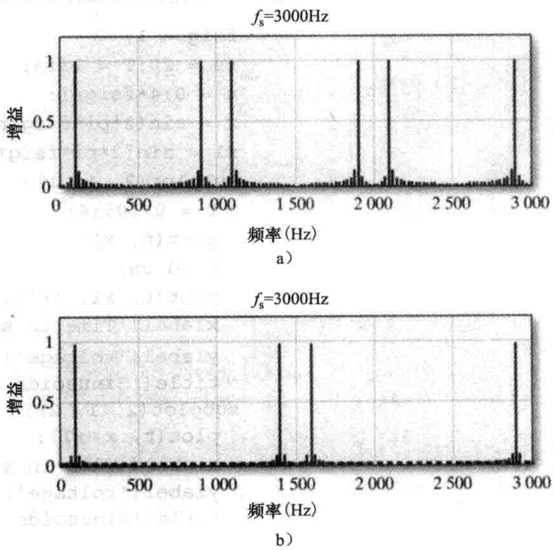


图 7.10 图 a)为以 1kHz 抽样后的 100Hz 信号，又升抽样为 3kHz 的频谱图，图 b)是以 1.5kHz 抽样后的 100Hz 信号，又升抽样为 3kHz 的频谱图，对于第二种情况，所需的低通滤波器的设计条件较松

例 7-4 内插滤波器的设计指标

信号的频率范围为 0～3kHz，抽样频率为 11025Hz。如果使用 10 位的 A/D 转换器，

且希望将信号升抽样到 22050Hz，如何给出合理的内插滤波器设计指标？

解： 以 11025Hz 抽样 0～3kHz 的信号，当对此抽样后信号进行升抽样至 22050Hz 时，会在 8025～11025Hz 产生镜像，需要将其滤除。由于 A/D 转换器是 10 位，可以将频带边缘限制为半位或更小(1 位对应 $1/2^{10}=0.000976$)，即要求 0～3kHz 通带上的增益为 1.0 ± 0.000488，8025Hz～$f_s/2$ 阻带上的增益为 0.000488 或衰减为 66dB，按上述条件，设计出的椭圆数字滤波器为 5 阶，帕克斯-麦克莱伦 FIR 数字滤波器为 17 阶。◀

级联内插器

与抽取器相似，当内插倍数较大(如大于 10)时，内插器通常也以级联方式实现。采用级联方式的原因可通过图 7.11 所示级联系统的频率响应来解释。U 倍升抽样在频域中产生 U 个镜像，其中只有第一个镜像为我们需要保留的信息，因而随着 U 值的增大，抗镜像滤波器的设计指标将会变得更加严格。级联两个或更多的内插器可以解决这个问题。

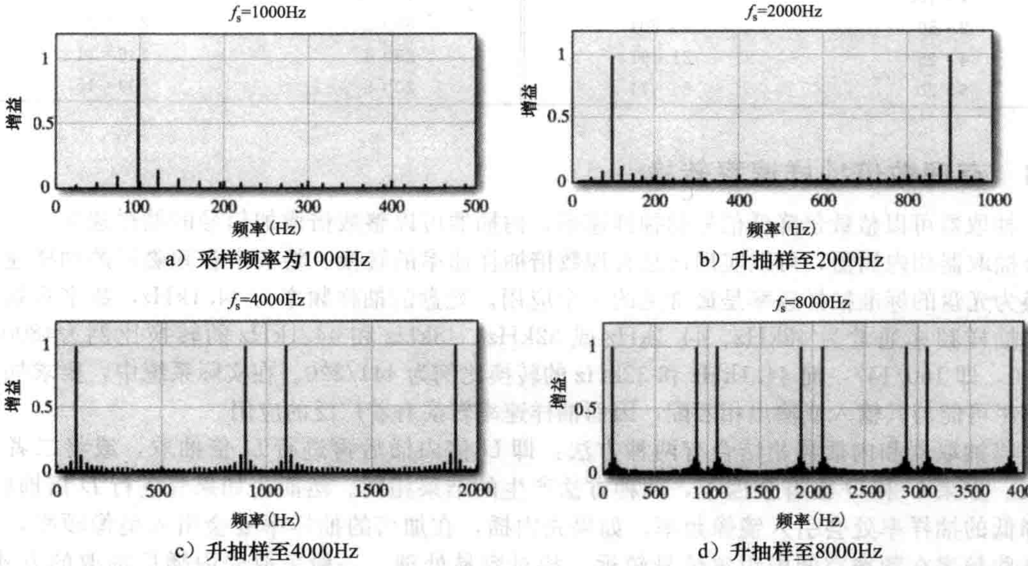


图 7.11　以 1000Hz 抽样的 100Hz 正弦信号频谱及其升抽样后的频谱

例 7-5　抗镜像滤波器设计指标

以 500Hz 对频率为 100Hz 的正弦信号抽样，现利用升抽样将抽样率升至 50000Hz，通带和阻带波动为 0.001，比较以 100 倍内插的单级内插器与两级级联内插器所需的抗镜像滤波器的差别。

解： 首先使用单级内插滤波器来升抽样，以 500Hz 对 100Hz 的正弦信号抽样，第一个镜像将出现在 $f_s-100=400$Hz 的地方，抗镜像滤波器的通带范围为 0～100Hz，通带波动为 0.001，阻带范围为 400～50000/2Hz，阻带波动为 0.001，根据该指标设计出的帕克斯-麦克莱伦 FIR 数字滤波器，其阶数为 543。

若使用两级级联内插滤波器来升抽样，100 倍内插有多种选择。为便于描述，选择两个 10 倍内插器。对于第一个抗镜像数字滤波器，其通带和阻带与 100 倍的内插器相同，但抽样频率由 50000Hz 变为 5000Hz，设计出的帕克斯-麦克莱伦 FIR 数字滤波器，其阶数为 54。

对于第二个抗镜像滤波器，抽样频率为 50000Hz，信号为以 5000Hz 抽样的 100Hz 正弦信号，在 4900Hz 的地方会出现混叠，因而第二个滤波器的通带范围为 0～100Hz，阻带范围为 4900～25000Hz，通带波动和阻带波动保持不变，设计出的帕克斯-麦克莱伦 FIR

数字滤波器，其阶数为33。

两个抗镜像滤波器的幅度响应如图7.12所示。

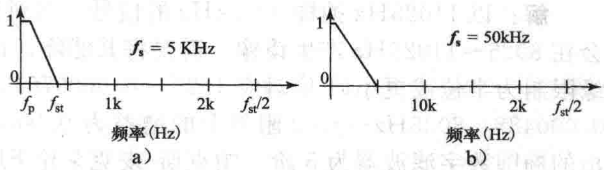


图7.12 两个抗镜像滤波器的幅度响应

对于100倍内插器，还有一些其他的级联组合，如2：50，4：25，5：20及它们的反序组合，同时也可使用其他方法设计低通数字滤波器，如IIR数字滤波器，其所需的阶数更低。表7.1给出了其他级联组合的帕克斯-麦克莱伦滤波器所需阶数。

表7.1 级联组合内插器和使用帕克斯-麦克莱伦低通滤波器设计抗镜像滤波器所需的阶数

级联组合	P-M阶数	级联组合	P-M阶数
1：100	543	10：10	54：33
2：50	8：204	50：2	272：2
4：25	21：91	25：4	136：11
5：20	26：71	20：5	109：15

7.3 有理数倍抽样速率转换

抽取器可以整数倍降低信号的抽样速率，内插器可以整数倍增加信号的抽样速率，因而结合抽取器和内插器，可以实现任意有理数倍抽样速率的转换。将数字音频磁带的抽样速率转换为光盘的标准抽样速率是最常见的一个应用。光盘的抽样频率为44.1kHz，数字音频磁带的抽样频率通常为48kHz、44.1kHz或32kHz。48kHz和44.1kHz的转换比例为48000/44100，即160/147，而44.1kHz和32kHz的转换比例为441/320。在实际系统中，要求抽样速率尽可能与其输入或输出相匹配，因而抽样速率转换有着广泛的应用。

将抽取器和内插器相结合有两种方法：即U倍内插后再进行D倍抽取，或将二者反过来。如果U和D没有公因子，两种方法产生的结果相同。然而，如果先进行D倍抽取，在降低的抽样率处会引入镜像频率。如果先内插，在加倍的抽样率处会引入镜像频率，因为这些频率在频率空间中距离信号较远，相对容易处理。一般采取先内插后抽取的方式，这样可以将抗混叠滤波器和抗镜像滤波器结合为一个滤波器，但滤波器的设计指标需要同时满足两者的设计指标。图7.13给出了二者的组合结构。

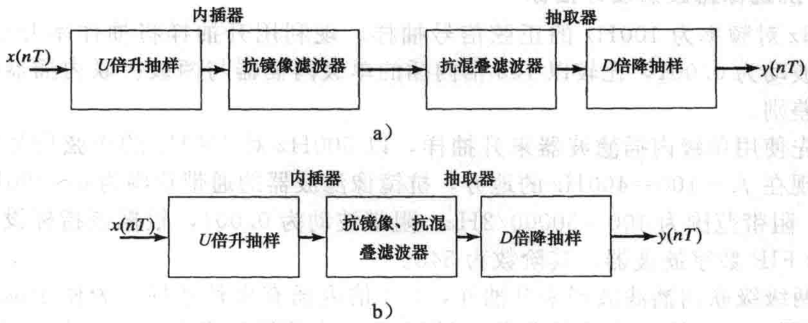


图7.13 结合内插器和抽取器实现有理数倍抽样速率转换

例7-6 抗混叠滤波器和抗镜像滤波器的选择

假设一带限信号的频率范围为0～100Hz，抽样频率为1000Hz，设计抽样速率转换器将抽样频率转换为2500Hz。

解： 2500和1000的公因子为500，因而内插和抽取比例为5/2，即5倍内插和2倍

抽取。

5 倍内插所需的抗镜像滤波器：抗镜像滤波器的抽样速率为 5×1000＝5kHz，原始信号的抽样频率为 1kHz，带宽为 100Hz，第一个镜像出现在 1000－100＝900Hz 处，因此通带为 0～100Hz，阻带为 900～2.5kHz。

2 倍抽取所需的抗混叠滤波器：抗混叠滤波器的输入信号抽样频率为 5kHz，以 2 倍抽取，将可能在 2.4～2.5kHz 之间引入混叠，因此通带为 0～100Hz，阻带为2.4～2.5kHz。

抗镜像滤波器的要求比抗混叠滤波器的要求更加严格，因此无须设计抗混叠滤波器，仅使用抗镜像滤波器即可。如果将通带波动和阻带波动设置为 0.001，使用帕克斯-麦克莱伦滤波器设计抗镜像滤波器，所需阶数为 19。图 7.14 给出了抽样速率转换过程中信号的时域和频域变化。

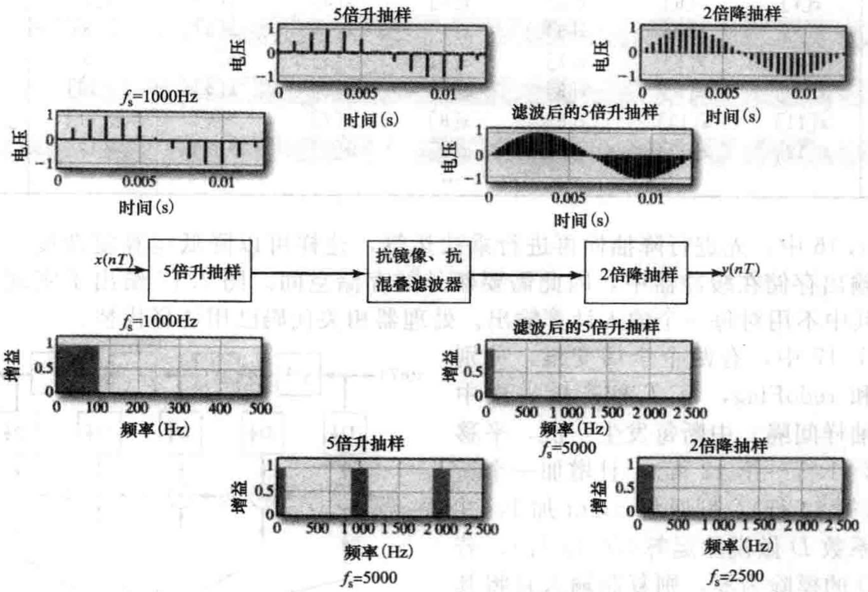


图 7.14　使用级联抽取器和内插器将抽样频率由 1000Hz 变为 2500Hz 的抽样速率转换系统，顶部为信号的时域变化，底部为信号的频域变化

7.4　FIR 数字滤波器的实现

抽取器由低通滤波器和将抽样频率降低 D 倍的降抽样器组成。降抽样器每 D 个样本仅输出一个样本信号，其余样本舍弃。因而，在低通滤波器中不对那些不被使用的抽样点进行运算，将提高运算效率。同样，内插器使用升抽样器在抽样点间插入 U 个零，并将结果通过内插滤波器。在输入为零的地方，内插滤波器无须做乘法运算，使用升抽样的性质也可以提高运算效率。本节将重点介绍内插器和抽取器中更有效的 FIR 滤波器实现方法。

7.4.1　抽取滤波器

FIR 滤波器的典型结构如图 7.15 所示，该结构实现了如下通用的差分方程：

$$y[n]=b_0x[n]+b_1x[n-1]+\cdots+b_Mx[n-M]$$
$$=\sum_{k=0}^{M}b_kx[n-k]$$

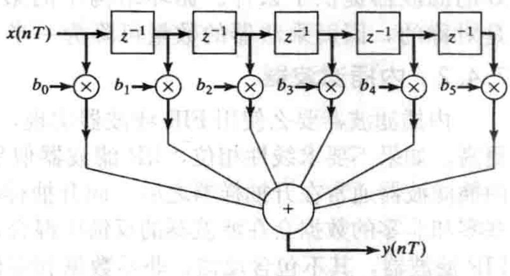


图 7.15　5 阶标准 FIR 滤波器结构

考虑 $D=2$ 的降抽样和 5 阶的 FIR 滤波器，表 7.2 给出了数据通过该滤波器的过程。

从该表可以看出，在 b_0 列，仅使用 $x[6]$，$x[8]$，$x[10]$，…，其为输入数据流的降抽样。同样，在 b_1 列中，使用的数据为 b_0 列数据的延时，即 $x[5]$，$x[7]$，$x[9]$，…，其为输入数据流的降抽样和延时。这可使用图 7.15 所示的结构实现。

表 7.2 该表显示了数据在 5 阶 FIR 滤波器中如何移动，最右端一列显示了 2 倍降抽样时，需要使用哪些数据[4]

时间	乘法系数						FIR 输出	降抽样
	b_0	b_1	b_2	b_3	b_4	b_5		
				…				
5	$x[5]$	$x[4]$	$x[3]$	$x[2]$	$x[1]$	$x[0]$	$y[5]$	—
6	$x[6]$	$x[5]$	$x[4]$	$x[3]$	$x[2]$	$x[1]$	$y[6]$	$y_D[3]$
7	$x[7]$	$x[6]$	$x[5]$	$x[4]$	$x[3]$	$x[2]$	$y[7]$	—
8	$x[8]$	$x[7]$	$x[6]$	$x[5]$	$x[4]$	$x[3]$	$y[8]$	$y_D[4]$
9	$x[9]$	$x[8]$	$x[7]$	$x[6]$	$x[5]$	$x[4]$	$y[9]$	—
10	$x[10]$	$x[9]$	$x[8]$	$x[7]$	$x[6]$	$x[5]$	$y[10]$	$y_D[5]$
11	$x[11]$	$x[10]$	$x[9]$	$x[8]$	$x[7]$	$x[6]$	$y[11]$	—
12	$x[12]$	$x[11]$	$x[10]$	$x[9]$	$x[8]$	$x[7]$	$y[12]$	$y_D[6]$
				…				

在图 7.16 中，先进行降抽样再进行乘法运算，这样可以降低运算复杂度，但要求降抽样器将输出存储在缓冲器中，因此需要额外的存储空间。图 7.17 给出了实现该算法的 C 代码，其中不用对每一个输入计算输出。处理器相关代码已用注释代替。

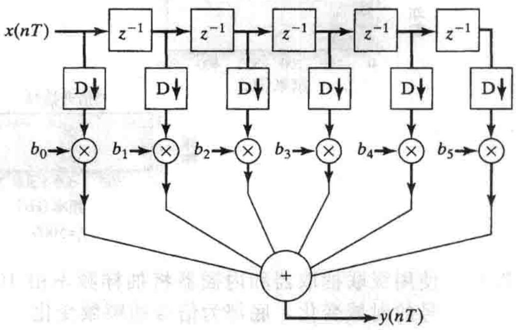


图 7.16 将降抽样器放在乘法器之前，以去除对结果没有贡献的抽样点

在图 7.17 中，有两个全局变量，分别为 count 和 redoFlag，每 T 秒产生一次中断，T 为抽样间隔。中断每发生一次，平移抽样点(第 41 行～第 42 行)，且增加一个新的抽样点(第 44 行)，同时将 count 加 1，并和降抽样系数 D 做模除运算(第 45 行)。若 count 和 D 的模除为零，则复制输入且将其存入缓冲器，同时将 redoFlag 置 1(第 46 行～第 49 行)。在主程序中，如果 redoFlag 为真，则将缓冲器中的值和滤波器系数相乘，并将其输出(第 26 行～第 33 行)。由于每 D 个抽样点进行一次乘法运算，则仅需每 TD 秒运行一次乘法运算。该结构中，对输入进行滤波的同时降低了抽样速率。计算效率方面，运算比较耗时的乘法器和累加器仅需在降低后的抽样速率上运行。通过增加缓冲器，获得了更多的时间来计算滤波器的输出，此为我们选择系数更多、对应的频率响应更好的滤波器提供了条件。循环结构中的第 29～30 行用于滤波器计算，由于滤波器的系数是对称的，因而乘法器的数量可降为一半。

7.4.2 内插滤波器

内插滤波器要么使用 FIR 滤波器实现，要么使用 IIR 滤波器实现，IIR 滤波器的计算效率更高。如果不要求线性相位，IIR 滤波器似乎是理想的选择，然而，实际情况并非如此。由于内插滤波器通常在升抽样器之后，而升抽样器会在抽样点间插入零，对于 IIR 滤波器来说，这些零和非零的数据会在滤波器的反馈中混合在一起，且没有一个很好的方法忽略这些零值。对 FIR 滤波器，其不包含反馈，非零数据和零值不会混合，可以设计出能够有效忽略零值的 FIR 滤波器结构。因此 FIR 滤波器有更好的计算效率：第一，零值可忽略，因而无须对这些项进行

```
1   /*FIRDown.c
2   This program does down-sampling by a factor of 2. It uses a 13th
3      order FIR Parks-McClellan filter as the low pass filter.
4  */
5  //Define the filter coefficients
6  const float b[] = {-0.01654471035, 0.03359069513, 0.04051543137,
7                    -0.05097669179, -0.08173283330, 0.14463751667,
8                     0.4492245433
9
10  const int N = 13;     //Filter order
11  int count, redoFlag;
12  double x[N+1];        //Input and past inputs
13
14  void Timer0ISR_irq(void);  //Define the ISR
15  int main(void)
16    {int i;
17     double y;
18     //Initialize the system clock
19     //Set up and initialize the A/D and D/A converter
20     //Set up Timer 0 to interrupt every T seconds
21     //Initialize adnn start the timer interrupt
22     //
23     count = 0;
24     redoFlag = 0;
25     while(1)
26       {if(redoFlag == 1)
27          {y = 0;
28           //This loop calculates the filter
29           for(i=0;i<(N+1)/2;i++)
30              y += b[i]*(x[i] + x[N - i]);
31
32           redoFlag = 0;
33
34       }
35    }
36  //
37  //This is the interrupt service routine.
38  void Timer 0ISR_irq(void)
39    {int i, val;
40     //Clear and restart the timer
41     for(i=N;i>0;i--)    //Shift the variables
42        x[i] = x[i-1];
43     //Start the A/D and wait for it to complete
44     // A/D returns in val
45     count = (count + 1) : 2;
46     if(count == 0)
47       {x[0] = val;
48        redoFlag = 1;
49       }
50     //Reset the interrupt
51    }
```

图 7.17 滤波器和降抽样器有效实现的伪代码

乘法运算和加法运算；第二，可以采用并行方式实现 FIR 滤波器。

为了便于理解，介绍一个简单的例子。假设内插滤波器之前有一个升抽样器，该升抽

样器在每两个抽样点间插入一个零，将抽样频率增加到原来的 2 倍。进一步假设滤波器的阶数为 5，长度为 6，系数分别为 b_0、b_1、b_2、b_3、b_4 和 b_5，如图 7.16 所示。输入到升抽样器的数据流为

$$x[n]=\{x[0],x[1],x[2],\cdots,x[n],\cdots\}$$

升抽样器在每两个抽样点间插入零，则输入到内插滤波器的数据流为

$$x_{\text{up}}[n]=\{x[0],0,x[1],0,x[2],0,x[3],0,\cdots,x[n],0,\cdots\}$$

表 7.3 给出了数据通过滤波器的乘法运算。

表 7.3 抽样点间插入零后，数据在图 7.15 所示的滤波器中的移动，表中最右侧一列为使用的系数

时间	乘法系数						FIR 输出
	b_0	b_1	b_2	b_3	b_4	b_5	
				…			
4	$x[2]$	0	$x[1]$	0	$x[0]$	0	b_0，b_2，b_4
5	0	$x[2]$	0	$x[1]$	0	$x[0]$	b_1，b_3，b_5
6	$x[3]$	0	$x[2]$	0	$x[1]$	0	b_0，b_2，b_4
7	0	$x[3]$	0	$x[2]$	0	$x[1]$	b_1，b_3，b_5
8	$x[4]$	0	$x[3]$	0	$x[2]$	0	b_0，b_2，b_4
9	0	$x[4]$	0	$x[3]$	0	$x[2]$	b_1，b_3，b_5
10	$x[5]$	0	$x[4]$	0	$x[3]$	0	b_0，b_2，b_4
11	0	$x[5]$	0	$x[4]$	0	$x[3]$	b_1，b_3，b_5
12	$x[6]$	0	$x[5]$	0	$x[4]$	0	b_0，b_2，b_4
13	0	$x[6]$	0	$x[5]$	0	$x[4]$	b_1，b_3，b_5
14	$x[7]$	0	$x[6]$	0	$x[5]$	0	b_0，b_2，b_4
15	0	$x[7]$	0	$x[6]$	0	$x[5]$	b_1，b_3，b_5
16	$x[8]$	0	$x[7]$	0	$x[6]$	0	b_0，b_2，b_4
17	0	$x[8]$	0	$x[7]$	0	$x[6]$	b_1，b_3，b_5
18	$x[9]$	0	$x[8]$	0	$x[7]$	0	b_0，b_2，b_4
				…			

从表 7.3 中可以清楚地看出，在偶数编号的时间周期中，仅对 b_0、b_2 和 b_4 系数进行乘法运算，在奇数编号的时间周期中，仅对 b_1、b_3 和 b_5 系数进行乘法运算。在任何时刻，都无须对 6 个系数同时进行乘法运算。这样，可认为该滤波器是两个近似独立的滤波器在不同时间周期中交替工作，且任何一个滤波器的长度明显小于完整滤波器的长度。图 7.18 给出了该结构。在该图中，两个滤波器的输入相同，但使用不同的系数来计算输出。复用器选择将哪一个滤波器的输出将作为真实的输出。除了使用相同的输入和使用相同的时钟以保证输出同步外，这两个滤波器相互独立，每一个滤波器均可使用独立的计算机系统或微处理器实现。

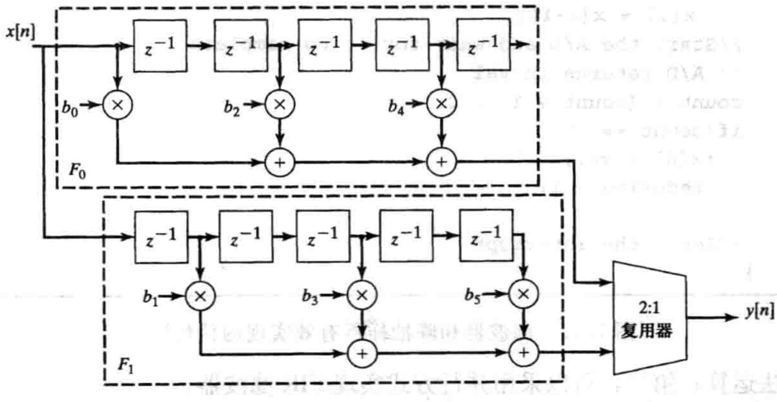


图 7.18 2 倍升抽样器后内插滤波器的有效实现原理图[5]

对于图 7.18 中所示的结构，时钟速率为升抽样速率，且每一个滤波器的输出均采用该速率进行计算，通过增加缓冲器存储延时的输入值，可以获得更多的计算时间。

表 7.3 所示为 2 倍升抽样器的数据，其可被扩展为更高倍数的升抽样器。对于 U 倍的升抽样器，可以得到 U 个独立的滤波器，每一次乘法运算所需的系数长度和滤波器的长度相关。在例 7-6 中，升抽样器系数为 5，Parks-McClellan 滤波器的阶数为 19，长度为 20，可使用 5 个独立的滤波器实现，每一个滤波器仅包含 4 个系数。

内插滤波器也可以通过软件实现，软件实现伪代码如图 7.19 所示。在图 7.19 中，主程序每 T 秒循环一次，但每个滤波器每 TU 秒计算一次，因此降低了乘法和加法运算次数。

内插滤波器也可以不用分解为多个独立滤波器的方式来实现，尽管我们可以获得其由于不对零值系数进行乘法运算所带来的益处。

```
Main program
Initialize variables
unsigned integer count = 0
Do forever
  x[0] = Get Next sample
  if count = 0
       y = calculate filter 0
  else if count = 1
       y = calculate filter 1
  else if count = 2
  ...
  output y to D/A
  shift variables for next cycle
    x[n] = x[n-1]
  count = count + 1 mod x
  wait for T seconds to pass
end of loop
```

图 7.19　利用升抽样器内插的补零有效实现内插滤波器的伪代码

例 7-7　内插升抽样滤波器设计

已知带限信号带宽为 200Hz，抽样频率 f_s 为 1000Hz，设计一内插升抽样滤波器，使抽样频率提高为 6000Hz，使用帕克斯-麦克莱伦设计内插滤波器，通带波动和阻带波动为 0.001，给出该滤波器实现的伪代码。

解：对于 200Hz 的信号，对抽样后信号进行 6000/1000=6 倍内插时，频域的第一个镜像出现在 $f_s-200\text{Hz}=800\text{Hz}$ 的位置，则通带为 0～200Hz，阻带为 800～3000Hz，通带和阻带波动为 0.001，设计出的帕克斯-麦克莱伦 FIR 数字滤波器阶数为 32(长度为 33)。使用 6 倍的升抽样且按照表 7.3 的方案，得到滤波器系数如表 7.4 所示(值在表 7.5 中)。该滤波器的幅频响应如图 7.20 所示，其 C 代码实现如图 7.21。

表 7.4　32 阶帕克斯-麦克莱伦滤波器中 6 个滤波器 F_0～F_5 及其对应的系数

滤波器	系数					
F_0	b_0	b_6	b_{12}	b_{18}	b_{24}	b_{30}
F_1	b_1	b_7	b_{13}	b_{19}	b_{25}	b_{31}
F_2	b_2	b_8	b_{14}	b_{20}	b_{26}	b_{32}
F_3	b_3	b_9	b_{15}	b_{21}	b_{27}	
F_4	b_4	b_{10}	b_{16}	b_{22}	b_{28}	
F_5	b_5	b_{11}	b_{17}	b_{23}	b_{29}	

表 7.5　32 阶帕克斯-麦克莱伦滤波器的系数

系数	值	系数	值
b_0 和 b_{32}	0.001384890316952	b_9 和 b_{23}	−0.016036782872540
b_1 和 b_{31}	0.002428494699234	b_{10} 和 b_{22}	−0.000904095936805
b_2 和 b_{30}	0.003327947399082	b_{11} 和 b_{21}	0.025268634136677
b_3 和 b_{29}	0.002960059531253	b_{12} 和 b_{20}	0.060239179319062
b_4 和 b_{28}	0.000390921038180	b_{13} 和 b_{19}	0.098867845332791
b_5 和 b_{27}	−0.004734090854961	b_{14} 和 b_{18}	0.134132972003924
b_6 和 b_{26}	−0.11634987865109	b_{15} 和 b_{17}	0.158880163206156
b_7 和 b_{25}	−0.018121238254999	b_{16}	0.167784990914451
b_8 和 b_{24}	−0.020829321732513		

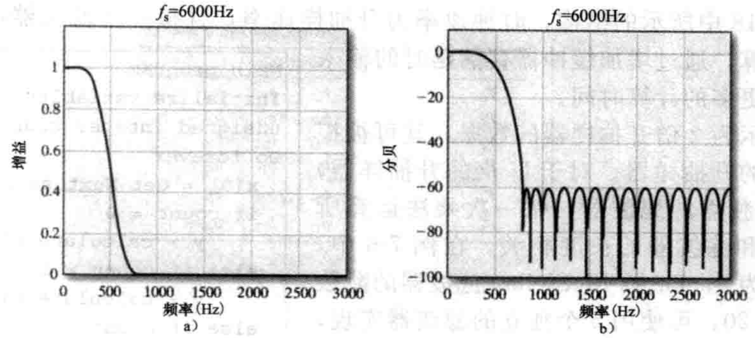


图 7.20　32 阶帕克斯-麦克莱伦滤波器的增益频率图

```
1  /*FIRUp.c
2  This is a FIR filter of order 32 which is used in an interpolator to
3    up-sample the input by a factor of 6.
4  */
5  //Define the filter coefficients
6  const float b[] = {0.00138489032, 0.00242849470, 0.00332794740,
7            0.00296005953, 0.00039092104, -0.00473409085, -0.01163498787,
8          -0.01812123825, -0.02082932173, -0.01603678287, -0.00090409594,
9            0.02526863414, 0.06023917932, 0.09886784533, 0.13413297200,
10           0.15888016321, 0.16778499091
11         };
12  const int N = 32;              //Order
13  const int U = 6;               //up-sample factor
14  int main(void)
15   {int i, count;
16     int val;
17     float y;
18     float x[N+1];        //input and past inputs
19     //Set up A/D and D/A converters
20     //Set up system clock
21     //Initialize timer and start it running
22     //Start the A/D Converter
23     while(1)
24     {//Wait for A/D converter to finish conversion
25      //  A/D returns answer in val
26      x[0] = val;
27      if (count == 0)
28        y = b[0]*x[0] + b[6]*x[6] + b[12]*x[12] +b[14]*x[18] + b[8]*x[24] +
                                     b[2]*x[30];
29      else if (count == 1)
30        y = b[1]*x[1] + b[7]*x[7] + b[13]*x[13] +b[13]*x[19] + b[7]*x[25] +
                                     b[1]*x[31];
31      else if (count == 2)
32        y = b[2]*x[2] + b[8]*x[8] + b[14]*x[14] + b[12]*x[20] + b[6]*x[26] +
                                     b[0]*x[32];
33      else if (count == 3)
34        y = b[3]*x[3] + b[9]*x[9] + b[15]*x[15] + b[11]*x[21] + b[5]*x[27];
35      else if (count == 4)
36        y = b[4]*x[4] + b[10]*x[10] + b[16]*x[16] + b[10]*x[22] + b[4]*x[28];
37      else if (count == 5)
38        y = b[5]*x[5] + b[11]*x[11] + b[15]*x[17] + b[9]*x[23] + b[3]*x[29];
39      //send y to the D/A converter
40      for(i=N;i>0;i--)      //Shift variable for next cycle
41        x[i] = x[i-1];
42      count = ((count + 1) % U);    //Increment count mod U
43      //Wait for timer to run out T seconds
44      //Restart A/D converter
45     }
46   }
```

图 7.21　32 阶帕克斯-麦克莱伦内插滤波器的 C 代码实现，处理器相关代码用注释代替

在图7.21中，第6～11行定义了滤波器系数，由于系数是对称的，仅需定义一半系数即可。第18行定义的数组 $x[N]$ 保存了当前和过去的输入值。第24～26行中的模数转换结果返回给变量 val。模数转换的结果在输入到滤波器前，常需要将其数据进行按比例地放大或缩小(缩放)，该过程此处不予以介绍。第27～38行是滤波器计算，每循环一次，count 变量加1后与6做模除。根据表7.4，“if”语句执行差分计算。滤波器的计算结果赋值给变量 y(y 也可能是缩放后的结果，该处没有显示)，且送往 D/A。第40～41行中，平移输入变量，为下一次循环做准备。在主循环的底部，将 count 加一后与6做模除，重启 A/D，等待抽样时间的结束。

7.5 窄带滤波器

现有某数字滤波器，其宽带相比于抽样频率来说非常小，通常来说，这样的数字滤波器需要高阶的 FIR 滤波器来实现。例如，抽样频率为1000Hz，通带为0～100Hz，阻带为150Hz～$f_s/2$，通带波动和阻带波动为0.01，帕克斯-麦克莱伦 FIR 设计出的数字滤波器所需的阶数为39。如果抽样频率变为10000Hz，滤波器的阶数将为389。这样的滤波器很难实现，同时在如此高的抽样频率下，可能没有足够的时间完成所有的乘法运算。解决这类问题的一个方案是使用抽取器将抽样频率降低，在降低后的抽样频率下对信号进行数字滤波，然后再通过内插器还原抽样频率，该原理图如图7.22所示。

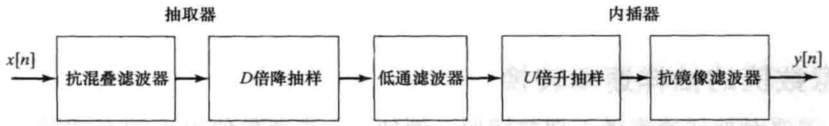


图7.22 在该图中，使用抽取器来降低抽样频率，这样可以在较低的速率上计算所需的低通滤波器，且滤波器的要求也会降低，然后使用内插器恢复初始的抽样频率

在上述解决方案中，可用三个滤波器(抗混叠数字滤波器、阶数相对较低的低通数字滤波器和抗镜像数字滤波器)替代一个高阶窄带的低通数字滤波器。不同于有理数倍抽样速率转换，此处抗混叠滤波器和抗镜像滤波器不能合二为一。

例7-8 抽取器-内插器的应用

以 $f_s=8000$Hz 抽样的信号需要通过通带为0～80Hz，阻带为160Hz～$f_s/2$ 的低通滤波器，要求通带波动为0.01，阻带波动为0.001，满足该要求的帕克斯-麦克莱伦滤波器的阶数为255，试使用抽取器和内插器将滤波器阶数降为60以下。

解： 使用内插器和抽取器存在一个不足，其将一个数字滤波器转变为三个数字滤波器实现，三个滤波器的通带增益相乘应为0.01，因此相比于使用一个滤波器，每一个滤波器的通带增益要求更加严格。如果三个滤波器中的任意滤波器的增益为 $1-r_p$，级联之后的总增益为 $(1-r_p)^3$，由于 r_p 的值很小，忽略 $(1-r_p)^3$ 中的高阶项，得到：

$$(1-r_p)^3 \approx 1-3r_p$$

因此，每一个滤波器的通带波动为0.01/3=0.0033。

使用如下 MATLAB 代码计算帕克斯-麦克莱伦低通数字滤波器的阶数。

```
fs = 8000;
fpass = 80;
fstop = 160;
rpass = 0.033;
rstop = 0.001;
F = ([fpass fstop]);
M = ([1 0]);
Dev = [rpass rstop];
[N F A W] = firpmord(F, M, Dev, fs);
disp(N);
```

为了使$N<60$，使用多个新的f_s值测试后，确定$f_s<1648$。如果令$D=5$，降低后的抽样频率$f_s=1600$，其对应的帕克斯-麦克莱伦滤波器的阶数为58，该过程见图7.23。

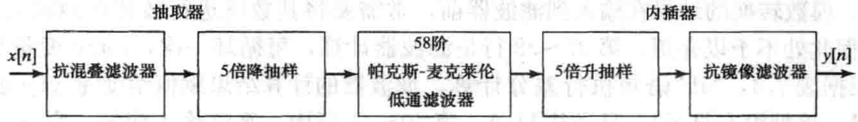


图7.23 使用降抽样来降低低通滤波器的阶数，然后使用升抽样还原抽样频率

如果抗混叠滤波器和抗镜像滤波器也使用帕克斯-麦克莱伦滤波器，其参数和所需的阶数见表7.6。

表7.6 抗混叠滤波器和抗镜像滤波器的参数

指标	抗混叠滤波器	抗镜像滤波器
通带	0～100Hz	0～100Hz
通带波动	0.0033	0.0033
阻带	800Hz～$f_s/2$	1500Hz～$f_s/2$
阻带波动	0.001	0.001
帕克斯-麦克莱伦滤波器阶数	32	15

7.6 任意数值的抽样速率转换

有时，需要对抽样速率做无理数转换，例如π，或者数值很大的有理数。对于这种情况，将采用与传统内插滤波器不同的内插方式。本节考虑两种近似的内插技术。第一个技术中，使用上一个抽样点值而不是使用零值作为抽样点间的内插值，我们称该技术为保持内插(hold interpolation)。另一种技术更符合直觉，称为线性内插(linear interpolation)，即在两个抽样点间画一条直线，内插值为该直线上的值。

7.6.1 保持内插

保持内插的算法见图7.24[2]，从第$n-1$个抽样点到第n个抽样点间画一条水平线，则新的信号变为连续信号，可使用任意的抽样速率重新进行抽样。

保持抽样器的一个最显著的优点就是很容易实现，其要求很少运算或者不需要运算。在软件实现时，第一个中断每T秒产生一次，得到$x[n]$的一个抽样值，并将其储存在一个缓冲器中，第二个中断每T_{new}秒产生一次，其仅将缓冲器中的值输出到$y[m]$即可。

保持抽样器的缺点为精度较低，图7.25比较了10倍保持内插器与理想内插器的频率响应。

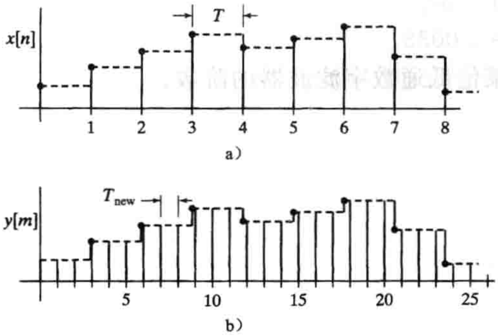


图7.24 原始信号$x[n]$被重新抽样为$y[m]$

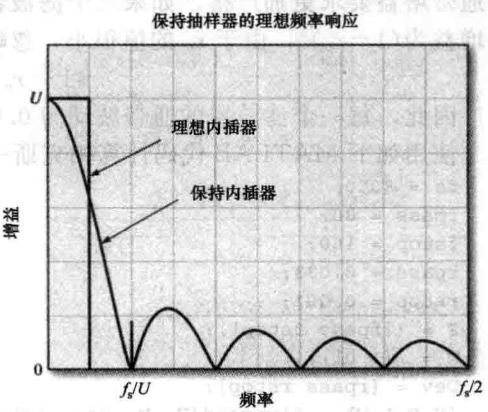


图7.25 10倍理想抽样器和保持抽样器的频率响应

7.6.2　线性内插

对于线性内插，在两个抽样点间画一条直线，将直线上的值作为新的内插值。在图 7.26 中，在信号 $x[n]$的第 n 个抽样点与第 $n-1$ 个抽样点间进行内插，内插后的抽样点值 x_s 为

$$x_s = x_n(t_s/T_0) + x_{n-1}(1 - t_s/T_0)$$

该公式中一个明显的问题是计算 t_s 时刻的抽样值需要知道 x_n 的值，然而 x_n 是未来时刻的值，如果 x 的值储存在存储器中，该问题可以解决，但在需要实时计算时，需要在缓冲器中存储足够多的 x 值，以输出上一个周期的 x_s 值。

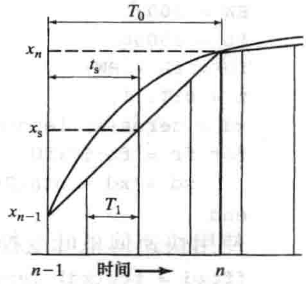


图 7.26　线性内插

5 倍线性内插滤波器的脉冲响应和频率响应如图 7.27 所示。

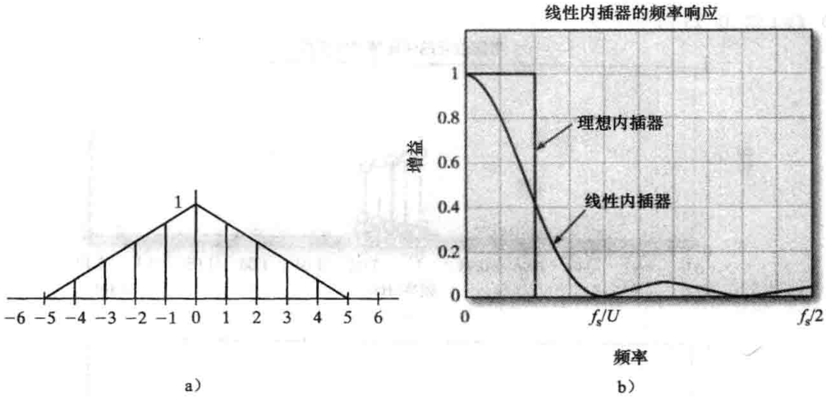


图 7.27　内插滤波器的脉冲响应和频率响应

7.7　带通抽样

带通抽样，又称为欠抽样(under-sampling)，是可用于改变抽样频率的另一种技术。当信号的带宽在某频带范围内，可以使用该技术改变抽样速率，如载波调制后的信号。在图 7.28中，对带通信号进行频率为 f_s 的抽样，将在抽样频率整数倍处产生原信号频谱的复制。如果带通信号的上下边频为 f_U 和 f_L，中心频率 $f_c=nf_s$，以 f_s 进行抽样，将得到完全相同的频谱。

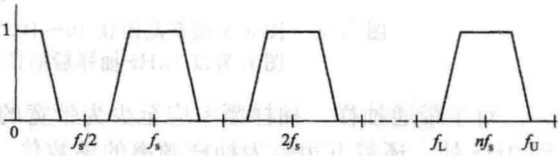


图 7.28　通带信号抽样后的频谱

这个概念可以在抽样频率小于两倍最高频率(即 $2f_U$)的情况下，恢复调制后的信号信息，例 7-9 给出了 MATLAB 实现过程。

例 7-9　带通信号的欠抽样

现有一单边带模拟信号如图 7.29 所示，其中 $f_L=10\text{kHz}$，$f_U=10.2\text{kHz}$，希望以 f_s 对该带通信号进行抽样，以使该信号的通带移到起始频率 0Hz。信号带宽为 200Hz，为保证恢复信号信息，抽样频率应大于 400Hz。同时要求将 f_L 移动到 0Hz 处，则 f_L 应为 f_s 的整数倍。为满足上面两个条件，选取 $f_s=1\text{kHz}$。

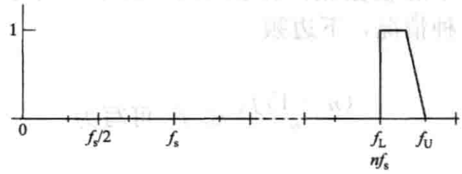


图 7.29　带通模拟信号

在 MATLAB 中，利用一组正弦信号之和来表示

带宽在 f_L 和 f_U 之间的信号，正弦信号的频率分别为 10kHz、10.05kHz、10.1kHz、10.15kHz 和 10.2kHz，抽样频率为 1kHz，如下：

```
fs = 1000;T = 1/fs;
BW = 200;
fL = 10000;
fU = fL + BW;
t = 0:T:.1;
xd = zeros(1, length(t));
for fr = fL:50:fU
    xd = xd + sin(2*pi*fr*t);
end
```

使用快速傅里叶变换(FFT)计算 1kHz 抽样后的带通信号频谱，结果如图 7.30 所示。

```
fftxd = fft(xd)/length(xd);
k = 0:length(t)-1;
stem(k*fs/(length(t)-1), abs(ftxd));
axis([0 fs1/2 0 1]);
```

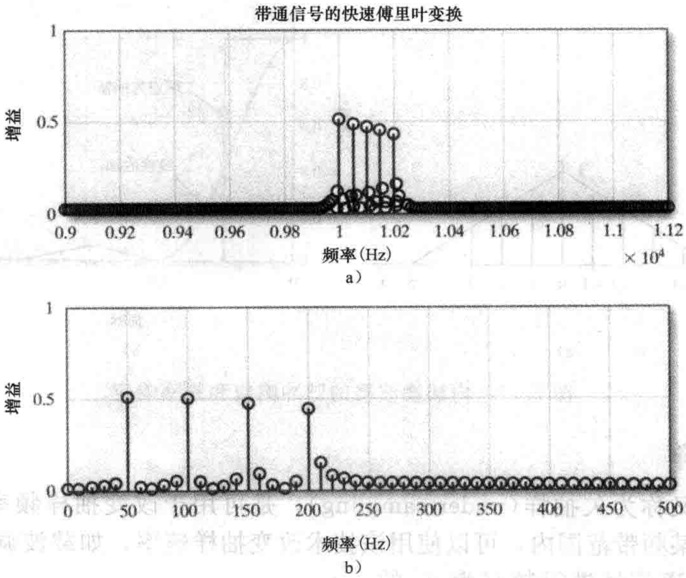


图 7.30　图 a 为频率范围在 10～10.2kHz 之间的原始信号频谱，图 b 为以 1kHz 抽样后的信号频谱

对于带通抽样，抽样频率应至少为带宽的两倍。此外，如果希望将信号的下边频移动到 0Hz 处，还需下边频为抽样频率的整数倍。

带通抽样也存在一些问题，欠抽样信号既包含原始信号的通带，也包含了空白频带，这些空白频带通常会包含噪声，导致信噪比下降。

一般来说，下边频通常不是抽样频率的整数倍，如图 7.31 所示，对于这种情况，下边频

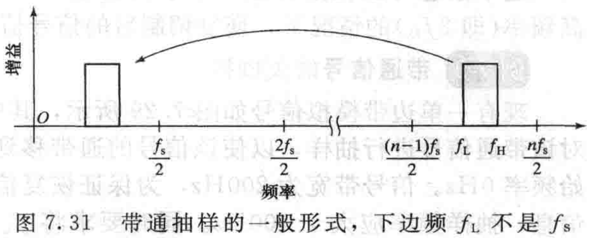


图 7.31　带通抽样的一般形式，下边频 f_L 不是 f_s 的整数倍

$\frac{(n-1)f_s}{2} \leqslant f_L$ 可写为

$$\frac{f_s}{2} \leqslant \frac{f_L}{(n-1)} \tag{7.1}$$

同样，对于上边频，$f_H \leqslant \frac{nf_s}{2}$ 可写为

$$\frac{f_s}{2} \geqslant \frac{f_H}{n} \tag{7.2}$$

组合式(7.1)和式(7.2)，得到带通抽样定理，表达如下：

$$\frac{2f_H}{n} \leqslant f_s \leqslant \frac{2f_L}{n-1}$$

其中，$n=2$，3，…，int(f_H/带宽)。

7.8　过抽样在音频中的应用

人耳听力的频率范围为 20～20kHz，数字音频系统的标准抽样频率为 44.1kHz。为了正确地记录高质量的数字音频，在 A/D 的前端需放置一个“砖墙”(brick wall)模拟抗混叠滤波器。在记录过程中，如果不通过该滤波器进行过滤，由任意元件产生的高于22050Hz 的噪声将会混入到信号中。在实际中，仅能得到截止频率为 22050Hz“砖墙”滤波器的近似形式，且价格昂贵。除此之外，受温度、湿度和老化的影响，模拟元件会发生变化。过抽样技术提供了一种替代模拟抗混叠滤波器的方法，且对滤波器的要求更宽松。当过抽样和 A/D 结合使用时，系统通常称为过抽样模数转换器。

为了更好地理解其工作原理，以 2 倍过抽样模数转换系统为例。希望该系统输出的数字信号抽样频率为 $f_s=44.1$kHz，因此过抽样系统的抽样频率应为 $2f_s=88.2$kHz。抽样频率提高了 2 倍，则通带范围变为 0Hz～$f_s/2$，阻带范围变为 $1.5f_s$ 到无穷，如图 7.32 所示。通带波动和阻带波动均为 0.01，满足上述条件的巴特沃斯滤波器的阶数为 6。

抗混叠滤波器的输出信号将通过频率为 $2f_s$ 的抽样保持电路，然后转换为数字序列。模数转换之后使用低通数字滤波器且将抽样速率降到 44.1kHz，图 7.33 给出了完整的过程。

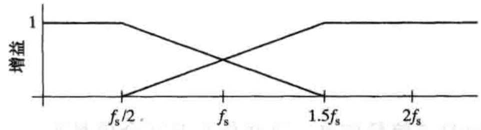


图 7.32　模拟滤波器的通带和阻带，如果通过该滤波器的模拟信号以 $2f_s$ 被抽样，则低于 $f_s/2$ 频率的信号不会产生混叠

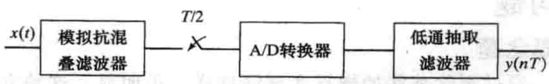


图 7.33　过抽样模数转换，将信号进行 2 倍升抽样，转换为数字序列，再通过抽取滤波器将抽样频率降低到 f_s 且阻止大于 $f_s/2$ 的频率通过

输出端的抽取滤波器不仅将抽样率降低为 f_s，而且是抗混叠滤波器，阻止大于 $f_s/2$ 的频率通过。更高倍数的升抽样将进一步降低对模拟滤波器的要求，但同时对模数转换的转换速率要求提高。

在音频系统的输出端，数模转换后的抗镜像滤波器也存在类似的问题。对于这种情况，可以在将信号输入到数模转换前对信号进行升抽样，降低对模拟抗镜像滤波器的要求和元件成本。

理想的抗镜像滤波器是一个低通滤波器，在 0Hz～$f_s/2$ 之间，增益为 1，其他频率处，增益为零。信号通过数模转换前，对其进行升抽样，可通过使用数字低通内插滤波器来降低对模拟抗镜像滤波器的要求，图 7.34 为其工作原理。在该图中，输入信号的抽样频率为 f_s，升抽样后再通过内插滤波器将镜像频率移到更高的频率范围，降低对模拟抗镜像滤波器的要求。

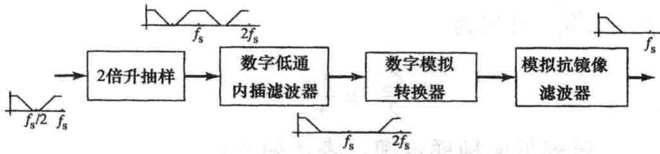


图 7.34 升抽样降低对模拟抗镜像滤波器的要求

小结

不同于其他章节，本章处理的信号为已抽样信号，观察抽样频率改变后信号的变化。降抽样过程是以某一规则删除某些抽样点以降低抽样频率，如果新的较低的抽样频率低于奈奎斯特抽样频率则会产生混叠。在大多数情况下，降抽样前使用低通滤波器去除高于奈奎斯特抽样频率的频率，低通抗混叠滤波器和降抽样器称为抽取器。

为了增加信号的抽样频率，可以在信号抽样点间插入零，这种方法的优点是增加了抽样频率，缺点是引入了镜像频谱。升抽样器和去除镜像频谱的低通滤波器称为内插器。

一般来说，对于范围较大的内插和抽取，采用级联的方式实现会相对容易，例如，采用两个 10 倍抽取的抽取器实现100 倍抽取，比直接实现 100 倍抽取容易实现一些。

抽取器和内插器均以整数倍数来改变抽样频率，将内插器和抽取器结合使用，可实现有理数倍的抽样速率转换。在这种情况下，可将抗镜像滤波器和抗混叠滤波器合并为一个数字滤波器。

通过考虑在某个特定周期中使用哪些滤波器系数，可以更加有效地实现内插器和抽取器。例如，内插器中很多抽样点的值为零，因此没有必要对这些零值做乘法运算，基于这个事实，可以改变滤波器的结构，大幅度提高效率。

实现任意数值的抽样速率转换也是可行的。为实现这个目的，需以某种近似方法重建模拟信号，对重建后的模拟信号进行重新抽样。保持内插和线性内插是实现重建模拟信号的两种方法。

习题

概念题

1. 通过删除某些抽样点实现降抽样，很明显，这种方法删除了信号信息，这些信息是冗余信息吗？该过程是否丢失了一些信息？
2. 当对信号升抽样时，通常在抽样点间插入零，然后再通过低通滤波器，这种方法是否创建了新的信息？如果是，其是否真实地反映了信号的信息？以 1kHz 抽样频率对范围在 0～100Hz 间的信号进行抽样和以 500Hz 抽样该信号后再使用内插器将抽样频率升高为 1kHz，两者结果有什么差别？
3. 相比于抽取前信号，列举抽取过程中引入的误差。
4. 相比于内插前信号，列举内插过程中引入的误差。
5. 在不要求线性相位的前提下，为什么不采用 IIR 滤波器实现内插滤波器？
6. N 级级联抽取滤波器，若每一个滤波器的通带波动为 R_{p1}，证明抽取后信号的波动 $R_p \approx NR_{p1}$。
7. 图 P7.7 中，信号 x 以 U 倍升抽样后再进行 D 倍降抽样，生成 y_1；x 以 D 倍降抽样后再进行 U 倍升抽样获得信号 y_2。设 x 为以 10Hz 抽样的频率为 1Hz 的正弦信号，绘制 x 的草图，并证明当 $U=D=2$ 时，y_1 和 y_2 不同，当 $U=3$，$D=2$ 时，y_1 和 y_2 相同。

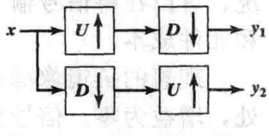


图 P7.7 升抽样和降抽样

8. 在第 5 章中，下面的公式给出了帕克斯-麦克莱伦 FIR 滤波器的阶数估计，在该公式中，ε_1 是通带波动，ε_2 为阻带波动，过渡带宽为 Δf，f_s 为抽样频率，使用下面的公式证明，对窄带信号滤波前进行降抽样是合理的。

$$N = \frac{-20\lg(\sqrt{\varepsilon_1\varepsilon_2})-13}{14.6\Delta f/f_s}$$

9. 设 100Hz 的正弦信号，抽样率为 1200Hz，希望通过保持内插技术，对该信号进行 3 倍升抽样，将抽样频率提高到 3600Hz。(a)应用保持内插后，估计信号的频率分量有什么变化；(b)使用 MATLAB 画出两个信号的 DFT，验证(a)中的估计。
10. 当采用升抽样和降抽样对抽样频率进行有理数倍转换时，一般先做升抽样，后做降抽样。如果先做降抽样后做升抽样，得到的结果相同吗？解释为什么。
11. 假设有一通过合适的抗混叠滤波器的方波，其谐波混叠已经不是问题，对该信号进行升抽样有什么困难？

分析设计题

7.1 节

7.1　信号包含两个正弦信号，频率分别为 1000Hz 和 3000Hz，抽样频率为 10000Hz。

(a) 如果按照每隔一点删除一抽样点的方法对信号进行 2 倍降抽样，估计信号频谱。

(b) 在 MATLAB 中产生两个正弦信号，使用 downsample 函数，验证(a)中的估计。

7.2　以 44100Hz 对信号抽样，该信号在 0～100Hz 范围内包含一些低频传感器数据，需要设计一个抽取器将抽样频率降为 11025Hz，同时要求传感器数据所处的频带范围波动不大于 0.01，设计该抽取器。利用 MATLAB 证明设计的抽取器满足上述要求，注意，混叠到通带中的阻带频谱不能引起通带波动超出设计指标。

7.3　下面的公式使用傅里叶序列拟合方波

$$x(t) = \sum_{\substack{k=1 \\ k\ 为奇数}}^{N} \frac{4}{k\pi}\sin(k\omega_0 t)$$

在附录 D 中，该公式利用 MATLAB 函数已实现。

```
function sq = fourier sq(fs, pts, f0, terms)
T = 1/fs;
n = 1:pts;
nT = n*T;
w = 2*f0*pi;
sq = zeros(1, length(nT));
for k = 1:2:2*terms
    sq = sq + (4/(k*pi))*sin(k*w*nT);
end
```

编写 MATLAB 代码调用该函数，其中 $f_s=8000$，pts=1024，$f_0=100$，terms=5。增加必要的代码，对该拟合的方波进行 D 倍降抽样，并画出结果。确定产生混叠信号的最小 D 值，描述当 D 增加时，由混叠现象引起的信号失真。

7.4　例 7-1 生成了频率范围为 10～2000Hz 的扫频信号，抽样频率为 11025Hz，当对信号进行 4 倍降抽样时，降抽样后信号的奈奎斯特频率变为 1378.1Hz，由于该频率低于信号的最高频率，出现了混叠并引起信号的严重失真。设计低通滤波器改善混叠问题，该滤波器满足下列设计指标：

抽样频率	22050Hz
通带	0～1178Hz
阻带	1578Hz～$f_s/2$
通带波动	0.02
阻带波动	0.01

在 4 倍降抽样前，利用 MATLAB 的 filter 函数，将原始信号通过该滤波器，画出下图：(a) 原始信号的时域图；(b) 原始信号的 FFT(将满足要求的巴特沃斯滤波器和椭圆滤波器的频域图与该图画在一张图上)；(c) 4 倍降抽样后信号的 FFT；(d) 使用巴特沃斯滤波器滤波后，降抽样信号的 FFT；(e) 使用椭圆滤波器滤波后，降抽样信号的 FFT。

7.5　在例 7-2 中，对信号进行 4 倍和 5 倍的降抽样，且使用通带和阻带如图 AP7.5 所示的两个低通滤波器，如果两个滤波器的通带波动和阻带波动均为 0.01，计算滤波器最终输出结果的通带波动和阻带波动。

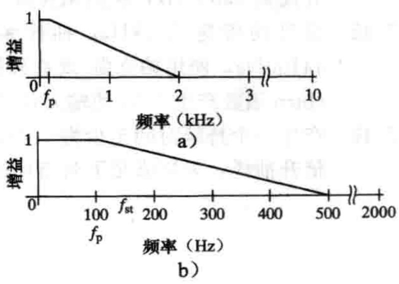


图 AP7.5　低通滤波器频带

7.6　现有一个信号，通带为 0～500Hz，噪声范围在 800～10000Hz，抽样频率为 22050Hz，希望对该信号进行 20 倍降抽样，为实现该目的，设计 4×5 的级联抽取器，若

通带波动和阻带波动均为 0.01，计算两个低通滤波器的通带和阻带范围。画出最终的降抽样信号，证明噪声被去除，且没有产生混叠。

输入信号创建：使用三角波的傅里叶级数展开，将求和项的频率限制在 500Hz 及以下。

$$\mathrm{tri}(nT)=\sum_{\substack{k=1\\ \text{奇数}}}^{5}\frac{8}{(k\pi)^2}\cos(200\pi nT)$$

增加噪声：使用方波的傅里叶级数展开，将求和项的频率限制在 800～10kHz。

$$\mathrm{sq}(nT)=0.2\sum_{\substack{k=1\\ \text{奇数}}}^{99}\frac{4}{k\pi}\sin(200\pi nT)$$

7.7 信号的通带为 0～100Hz，抽样频率为 40000Hz，希望将抽样频率降到 1000Hz，即 $D=40$。要求抽取器的通带波动 r_p 为 0.01，阻带波动 r_s 为 0.001，如果该抽取器按照 2-4-5 的级联方式设计，给出使用 Parks-McClellan 设计该滤波器的指标要求。

7.8 图 AP7.8 给出了将抽样频率降低 D 倍的抽取器，证明输出结果 y 可以写成 $y[n]=\sum_{i=0}^{M}b_i x[nD-i]$，其中，$b_i$ 是抽取滤波器的系数[3]。

7.9 三级级联抽取器，每一级滤波器的通带波动为 r_p，则滤波器最终的通带波动是多少？证明滤波器最终的通带波动约为 $3r_p$。

7.2 节

7.10 MATLAB 中的函数 upsample 通过在信号抽样点间内插零提高抽样频率，假设正弦信号的频率为 1000Hz，抽样频率为 5000Hz，估计使用 upsample 函数后信号的频谱。(a)2 倍升抽样；(b)3 倍升抽样，利用 MATLAB 验证你的估计。

7.11 假设正弦信号的频率为 f_{sig}，对其进行 2 倍升抽样。

(a) 证明升抽样后的信号频谱与频率为 $f_{sig}+(f_s-f_{sig})$ 的正弦信号的频谱是相同的，其中 f_s 为原抽样频率。

(b) 证明 3 倍升抽样后的信号频谱与频率为 $[f_{sig}+(f_s\pm f_{sig})]$ 的正弦信号频谱是相同的。

7.12 图 AP7.12 所示是将抽样频率增加 U 倍的内插器，证明输出结果 y 可以写成 $y[n]=\sum_{i=0}^{M}b_i x\left(\frac{n-i}{U}\right)\delta[n-i]$，其中，$b_i$ 是内插滤波器的系数[3]。

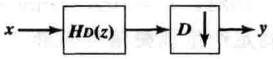


图 AP7.8 抽取滤波器和降抽样器

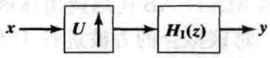


图 AP7.12 升抽样器和内插滤波器

7.13 假设有两个完全相同的信号，其为 1000Hz 附近的窄带信号，第一个信号的抽样频率为 11025Hz，第二个信号的抽样频率为 22050Hz。如果将两个信号均升抽样为 44100Hz，两个信号的频谱特征有什么差别？

7.14 信号的带宽为 0～4kHz，抽样频率为 11025Hz，如果使用 8 位模数转换器，同时希望将抽样频率升高到 22050Hz，使用帕克斯-麦克莱伦数字滤波器，设计一个合理的内插滤波器。

7.15 信号的带宽为 2kHz，抽样频率为 11025Hz，设计一内插升抽样滤波器将抽样频率升高为 44100Hz，使用帕克斯-麦克莱伦数字滤波器，通带波动和阻带波动为 0.001，利用 MATLAB 的 chirp 函数产生 0.9s 的输入信号，其幅度为 1，频率范围为 10～2kHz。

7.16 产生一个持续时间至少为 0.4s，频率为 80Hz 的正弦信号，抽样频率为 500Hz，对该信号进行 2 倍升抽样，采用满足下列指标的帕克斯-麦克莱伦内插滤波器，画出一个周期内的结果。

通带	0～110Hz
通带波动	0.01
阻带	250～500Hz
阻带波动	0.01

创建另一个信号，其为原始信号的 2 倍升抽样，但使用两个抽样点的平均值而不是零值进行内插，画出一个周期的结果，将其与内插滤波器的输出结果进行比较。

7.3 节

7.17 用以下倍数转换抽样速率，找到满足该条件的最小的内插器/抽取器的整数值。

(a) 0.65625000

(b) 0.52

(c) 2.375

7.18 假设某带限信号的频率范围为 0～400Hz，抽样频率为 1500Hz，设计一有理数倍的抽样速率转换器将抽样频率增加到 3500Hz，若要求通带和阻带波动为 0.001，使用帕克斯-麦克莱伦方法设计该数字滤波器，则该滤波器的通带截频和阻带截频分别是多少？使用如下输入信号和 MATLAB 中的函数 filter 验证上述结果。

```
fs = 1500;T = 1/fs;
Pts = 1024;
t = 0:T:(Pts - 1)*T;
s = chirp(t, 0, Pts*T, 100);
```

7.19 假设某带限信号的频率范围为 0～100Hz，抽样频率为 500Hz，设计一个有理数倍的转换器将抽样频率增加到 800Hz，要求通带和阻带波动为 0.001，利用下面 MATLAB 代码产生的信号测试该系统，并验证该抽样频率转换器将抽样频率变为 800Hz。

```
fs = 500;T = 1/fs;
Pts = 1024;
t = 0:T:(Pts - 1)*T;
s = chirp(t, 0, 1, 100);
```

7.4 节

7.20 现有一带限信号，其频率范围为 0～100Hz，以 500Hz 对该信号进行抽样，要求对该信号进行 12 倍的升抽样，

(a) 设计一个内插滤波器，其通带波动和阻带波动均为 0.01，使用帕克斯-麦克莱伦设计滤波器。

(b) 按 3×4 的级联方式设计该内插滤波器，重复上述问题。

7.21 图 7.17 给出了设计降抽样器的 C 代码，该代码中的每一行前面都有参考编号，回答下面的问题：

(a) redoFlag 变量的作用是什么？

(b) 第 30 行用于滤波器计算，为什么该方程写成

```
y += b[i]*(x[i] + x[N-i])
```

而不是

```
y += b[i]*x[i]
```

(c) 第 45～46 行中 count 变量的作用是什么？

(d) 中断服务程序多长时间发生一次？以抽样时间表示。

(e) 解释为什么在第 10 行中滤波器的阶数设为 13，而在第 6～8 行中，仅使用 7 个参数。

(f) 该代码为 2 倍降抽样，如变为 3 倍降抽样，程序的哪些地方需要改动？

(g) 在新抽样点准备好前，第 29～30 行的主循环有多长时间完成运算？

7.22 将升抽样倍数由 2 变为 3，重画表 7.2 和图 7.16。

7.23 设计一个内插器，其使用线性内插将输入信号的抽样率提高 5 倍，利用如下 MATLAB 代码产生输入信号：

```
fs = 1500;T = 1/fs;
n = 0:1/T;
nT = n*T;
x = chirp(nT, 10, 0.2, 500);
```

注意，该程序中的 chirp 函数产生的信号在 0.2s 时刻达到 500Hz，但该函数将运行到 1s，因而信号的频率持续增加。当频率高于 $f_s/2$ 时，将产生混叠。使用线性内插对信号进行 5 倍升抽样，我们希望原始信号与升抽样后信号的样点数相同，因而升抽样后信号时间范围为 0.2s，比较时间范围在 0～0.2s 内的原始信号与升抽样后信号。相比于传统的低通内插滤波器，使用线性内插的升抽样技术是否更有优势？

7.5 节

7.24 现有一个抽样频率为 11025Hz 的信号，需将其通过通带为 0～100Hz，阻带为 200Hz～$f_s/2$ 的低通滤波器，要求通带波动为 0.01，阻带波动为 0.001，参考例 7-8，设计一个抽取器/内插器结构。

7.6 节

7.25 设带限音频信号的带宽为 3kHz，抽样频率为 8kHz，通过升抽样将抽样频率提高为 11025Hz，以(a)保持内插和(b)线性内插做升抽样，利用如下 MATLAB 代码产生输入信号：

```
fs = 8000;T = 1/fs;
n = 0:4096;
nT = n*T;
x = chirp(nT, 10, nT(length(nT)), 3000);
```

比较在 0.005s 时间范围内，保持内插信号与线性内插信号的区别，注意保持内插中的阶梯效应。

7.7 节

7.26 如下 MATLAB 代码创建了一个调频信号，其频率范围为 40～40.25kHz，抽样率为 2kHz，增加必要的代码，画出该信号的奈奎斯特频率范围。

```
fs = 2000;T = 1/fs;
fL = 40000;
fU = 40250;
t = 0:T:3;
x = chirp(t, fL, 3, fU);
```

7.27 如下 MATLAB 代码创建了一个调频信号，其频率范围为 20000～20050Hz，抽样率为 80kHz，在频带范围为 1100～19000Hz 处加入振幅为 0.01 的噪声，以 f_s=1000Hz 降抽样该信号，证明降抽样后，带外噪声变为带内噪声，画出加噪声后的信号和降抽样后信号的 FFT 图。

```
fsh = 80000;Tsh = 1/fsh;
fL = 20000;
fU = 20050;
tsh = 0:Tsh:.5;
x = chirp(tsh, fL, .5, fU); %signal
%Add in noise
xn = .01*chirp(tsh, 1100, .5, 19000); %noise
x = x + xn;
```

7.28 如图 AP7.28 所示的单边带信号，设 f_L=50kHz，f_U=50.3kHz，如果以 8kHz 对该信号抽样，该信号被搬秒到 0Hz～$f_s/2$ 之间的哪个频率范围？多大的抽样频率可以将 f_L 搬移到 0Hz 处？

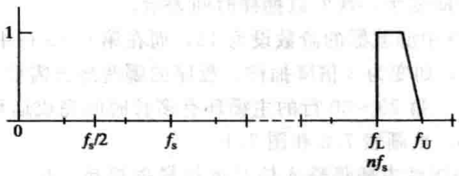


图 AP7.28 单边带信号

第8章 数字滤波器的实现

由设计指标求解传输函数的数字滤波器设计是一个易于理解的过程，根据设计指标，可以设计出一个优化的(某种准则下)IIR 或者 FIR 滤波器。但是，如果根据传输函数去获得一个最优的实现，这个过程往往较为复杂，并且可能仅是特定技术条件下的最优。换言之，今天通过优化设计获得的传输函数在将来可能仍然是一个优化的传输函数，但对于当前最优的实现未必适用于将来。

本章假定我们已经完成了数字滤波器的设计，将重点讨论滤波器的实现问题。滤波器设计完成，意味着我们已有该滤波器的传输函数，并且该传输函数的响应满足特定的设计要求。滤波器的实现分为两个层面，一是模型实现(Realization，以后简称实现)，典型的表示形式是传输函数对应的方框图或者信号流图。二是工程实现(Implementation，以后简称实施)，其按照模型实现给定的形式生成一个“真实”的滤波器，该真实的滤波器接收输入并产生对应的输出。实施过程中不可避免地将在真实滤波器与传输函数描述的滤波器之间产生误差。

由相同的传输函数可以派生出多种不同的实现与实施，假如利用无限精度运算且计算速度不受限制，则多种变化形式将通向相同的滤波器。某些实现对误差比较不敏感，某些计算效率较高，这就导致了在实施过程中对于相同的传输函数却产生多种不同的滤波器。

首先，我们将从滤波器实施的描述方式开始，包括描述抽样值以及从框图描述传输函数如何产生输出。然后，讨论实施过程中出现的主要误差，包括量化误差和舍入误差。最后，研究说明软件实现过程，并给出一些 C 语言的实例。

8.1 实施问题

实施的广义定义包括了信号抽样过程，将信号由电压转变为数字流的 A/D，计算机(或者数字信号处理(DSP)芯片)，运行滤波器算法的软件，以及最终将计算机的输出转变为连续时间信号的 D/A 和重建电路。沿着这条由输入到输出的路径，存在着多处产生误差的地方。某些误差无法避免，以至于实施所得到的滤波器仅是传输函数的近似，所以，滤波器的实施过程是在物理限制条件下的误差最小化。最基本的限制是成本和速度，其他的限制包括能耗限制、空间限制以及使用软件和硬件带来的限制。实施过程中产生的误差可以归纳为以下几类。

1. 输入量化误差：数字滤波器的输入信号一般是连续时间信号，需要经过 A/D 抽样后输入系统。A/D 抽样将连续时间信号转换成有限长的二进制数，所以，量化后的输入信号是连续时间抽样值的近似。

2. 混叠误差：就其本质而言，连续时间信号是非带限信号，任何高于奈奎斯特频率的频率分量都将与低于奈奎斯特频率的频率分量相混叠。

3. 系数量化误差：与输入量化相似，滤波器的系数也存在量化，原因来自利用有限长位数表示滤波器的每个系数。所以，所实施的传输函数仅是所设计的传输函数的近似。

4. 乘积量化误差：乘积运算中，两个 n 位的数字相乘结果应为 $2n$ 位，乘积运算需调整位数以适应寄存器，从而导致误差。

5. 溢出误差：当一个数字过大而超出系统位数表示的范围时，就会产生溢出误差，

经常发生于输入信号的 A/D 抽样，或者运算过程中。

6. 重建误差：将离散系统的输出重建为连续时间、连续幅度的模拟信号时，将产生重建误差。

系统输入量化需在硬件级处理，滤波器设计者需要选择抽样器和 B 位 A/D 转换器以适合应用的需要。在一些应用中至少需要 6 位描述信号，如视频信号的灰度级；另一些应用中，有可能需要利用 10～12 位，如语音信号处理；某些应用可能需要 14～20 位，如用于音乐的 CD 通常利用 16 位记录数字音频。在涉及计算机计算时，双精度浮点数可用 64 位表示，16 位表示的信号看起来似乎是一个很小的数字。然而，考虑到 16 位的数字表示了 65536 的分级，所以一个 16 位表示意味着我们已知这个信号的 5 位(十进制)有效数字，12 位表示了 4096 的分级，意味着已知近似 3 个半有效数字。对于大多数类型的数据，获得 16 位精度的测量值是十分困难的，常常超过 10 位将变成无效。例如，温度、压力、湿度以及亮度的 8 位测量值对大多数消费类产品已经足够。

混叠误差可通过在 DSP 系统前端放置抗混叠滤波器来处理。理想的抗混叠滤波器将所有高于系统奈奎斯特速率的频率增益降低至小于量化误差(最低有效位的 1/2)。实际应用中，通过过抽样滤波器或者限制有效频率范围低于奈奎斯特速率的方式，有望达到理想的效果。

系数量化误差、乘积误差和溢出误差需要在硬件选型后进行处理，硬件肯定会影响这三种误差。

其他必须面对的设计问题包括抽样速率的选择(直接与量化误差相关)、滤波器类型(IIR、FIR 或者其他算法)、系统数量以及滤波器算法运行精度问题等。

8.2 数字表示

一个 B 位二进制数可以表示为

$$\text{2 的幂次：}\quad 2^{B-1}2^{B-2}2^{B-3}\cdots 2^2 2^1 2^0$$

$$\text{二进制位：}\quad b_{B-1}b_{B-2}b_{B-3}\cdots b_2 b_1 b_0$$

其中，b 表示"1"或者"0"。

这个数字可以写为如下形式

$$n_o = \sum_{i=0}^{B-1} b_i \cdot 2^i \tag{8.1}$$

式中，n_o 是数字 n 的量化形式。将 n 和 n_o 归一化至 0～1 的范围内，随着 B 的增大，n_o 值越近似 n 真实值。

式(8.1)称为二进制形式或者正的二进制形式，其他建立在这个基础上的二进制形式使得我们可以表示有符号数和浮点数。

8.2.1 二进制补码表示

在 DSP 应用中，我们经常处理缩放的、有符号的二进制数，例如，A/D 转换器将范围为－10～＋10V 的连续信号转变成 10 位二进制数。二进制数可以表示－1～1 之间的实数，利用二进制数与比例因子乘积表示其他范围的数值。二进制补码系统可用来表示正数和负数。

对于 B 位的二进制补码系统，二进制数的一半用于表示正数，而另一半用来表示负数。正数的表示方式与二进制表示相同，负数的补码就是对其对应的正数取反码再加 1。所以，在 8 位二进制补码系统中，－43 的表示可以从＋43 的二进制表示开始。

＋43＝	00101011
取反→	11010100
加 1→	11010101＝－43

该系统中，对于正数，1 承载了数字的“权重”，添加更多的 1 可增大正数的值，而正数前面补零不改变正数的值。对于负数而言，情况恰好相反，所以 −43 的 16 位表示可以写为 1111 1111 1101 0101。

表 8.1 给出了 4 位系统的二进制补码。

表 8.1　二进制补码系统中的 4 位二进制数

n	二进制补码	*n*	二进制补码	*n*	二进制补码	*n*	二进制补码
+7	0111	+3	0011	−1	1111	−5	1011
+6	0110	+2	0010	−2	1110	−6	1010
+5	0101	+1	0001	−3	1101	−7	1001
+4	0100	0	0000	−4	1100	−8	1000

利用 4 位可以表示 −8～+7 之间的整数，最高位 0 代表正数，1 代表负数。所以，最高位称为符号位。

二进制补码系统的两个重要特征，其在 DSP 应用中十分有用。

1. 正数加上负数相当于减法。

2. 对于结果处于表示范围内的情况，中间溢出不影响结果精度。

第一个特征似乎是我们对于任何数字表示所期望的，但不完全是这样，例如一些其他的表示，包括二进制反码(1’s Complement)表示法[⊖]和符号/数值(Sign and Magnitude)表示法。为了说明第二个特征，这里考虑如下问题，4 位二进制补码系统如表 8-1 所示。

4	0100	
+7	0111	
=11	1011	中间溢出
−5	1011	
=6	10110→0110	4 位正确结果

该问题中，+4 加 +7 结果为 11，这将无法在 4 位二级制补码系统中表示，结果 1011 在二进制补码系统中表示 −5，如果在这里结束运算，将产生溢出误差。如果在该中间结果上加 −5，得到答案为 10110，经截尾至 4 位生成正确结果。换言之，一个二进制补码系统在正确的设计输入输出表示中，中间溢出误差对处理结果将不会产生影响。

二进制补码可以写为

$$r_o = A\left(-b_{B-1} + \sum_{i=0}^{B-2} b_i 2^{-i}\right)$$

其中圆括号中的部分被归一化至 $-1 \leqslant x < +1$ 的范围内。

这个公式中，r_o 表示实数 r 的量化值，常数 A 表示幅度或者缩放因子。对于接受输入电压范围为 ±10V 的 A/D 来说，这里的 A 为 10。B 是式中二进制位的数目，b_{B-1} 为最高有效位(符号位)。

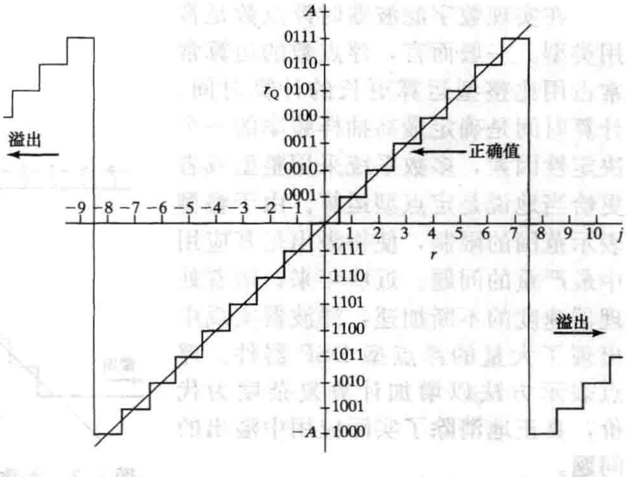


图 8.1　二进制补码计数映射图，r 表示实数，r_Q 表示量化的二进制补码，A 为缩放因子

图 8.1 给出了实数到二进制补码

⊖ 二进制反码用正数取反来表示负数，这种方法的 0 有两种表示方式。

计数系统的映射图，其中横坐标表示实数，纵坐标表示对应实数的离散(量化)的二进制补码数字。

使用 B 位可以表示 2^B 个离散电压值，若这些电压的范围是 $-A\sim+A$，则最小的电压表示为

$$Q = 2A/2^B = A/2^{B-1}$$

例如，若 A 取 10V，B 为 12 位，则最小量化值即为 $Q=20/4096=4.8828$mV。

图 8.1 中，r 和 r_Q 之间的 45°直线为 r 的真实值，所以量化产生了 $\pm Q/2$ 的误差。

8.2.2 符号/数值表示

二进制补码表示法是目前常用的一种方法，但也还有其他的表示方法。符号/数值表示法就是其中的一种，通过在正的二进制值表示前面增加 1 位符号位构成，如图 8.2 所示。

与二进制补码计数系统相同，符号位 1 表示负数，0 表示正数。例如，在 8 位数值系统中，数字+9 可以表示为符号/数值形式+9→00001001，而−9 可以表示为−9→10001001，其中最高有效位为符号位。在此数值系统中，对于数字 0 有两种表示方式：+0→00000000 和−0→10000000。

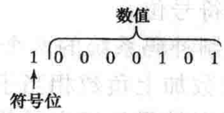


图 8.2 量化的二进制数的符号/数值表示法，数字−5 由数值+5 和一个负的符号位组成

在符号/数值表示法中，对于一个正数加上一个负数不等同于相减，为了实现正数和负数的加法运算，可以通过从正数上减掉对应负数的数值后调整结果的符号实现。B 位符号/数值表示法可以表示的数值范围为 $\pm 2^{B-1}$。所以 8 位数字系统可以表示−127～+127 范围的数字。(二进制补码系统中 0 的表示只有一种表示方式，所以可以表示的范围为−128～+127。)

对于符号/数值表示法，经常使用到的是饱和溢出。饱和溢出中，可以将所有溢出值设置为可以表示的最高值。例如，8 位数值系统中，所有大于+127 的数都将映射到+127，所有小于−127 的数映射到−127，图 8.3 给出说明。

8.2.3 浮点表示

在实现数字滤波器时浮点数是常用类型。一般而言，浮点数的运算常常占用比整型运算更长的计算时间。计算时间是确定最高抽样频率的一个决定性因素，多数系统采用整型或者更恰当地说是定点型运算。由于整型表示范围的限制，使得溢出是其应用中最严重的问题。近些年来，随着处理器速度的不断加速，滤波器实现中出现了大量的浮点型 DSP 器件。浮点表示方法以增加计算复杂度为代价，真正地消除了实际应用中溢出的问题。

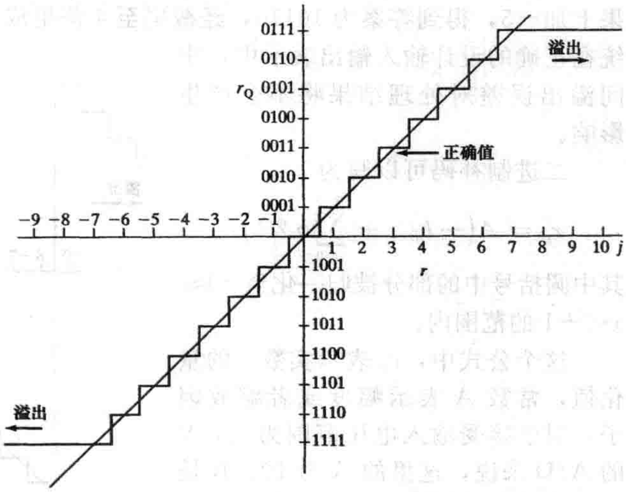


图 8.3 实数的符号/数值表示，具有饱和溢出功能，该图为 4 位计数系统

为了表示一个浮点数，需要同时表示其尾数部分和指数部分。计算机应用中从 20 世纪 50 年代开始使用浮点数，存在多种不同格式的表示方法。计算机制造商选择合适的浮点数格式去适用特定的应用程序和计算机结构。通过逐渐演变得到一种标准

的表示格式，绝大多数支持浮点运算的 DSP 器件遵守 IEEE 754/854 标准。该标准同时支持单精度(32 位)和双精度(64 位)格式，表示方法如图 8.4 所示。



图 8.4　IEEE 标准表示方法

对于图 8.4 中单精度格式，S 为符号位，“0”表示正尾数，“1”表示负尾数。尾数部分归一化至 0～1 的范围，归一化表示尾数的最高有效位始终为 1。也就是说，执行运算时平移尾数部分使得最高有效位为 1，并且对指数部分作相应调整。由于最高有效位始终为 1，因此不需要对它进行记录，该位将被省略。M 表示尾数，E 表示指数，数字 F 可表示为下面的格式

$$F=(-1)^{S}1.M\times 2^{E-127}$$

注意，指数部分 E 偏移一个因子 127。所以，真实的指数在二进制补码中处于 −127～+128 之间。通过指数真实值加 127 可以获得偏置指数。指数部分为 8 位，可以表示 256 个值。其中的两个值(全 1 和全 0)用于保留表示特殊的数，如表 8.2 所示。

双精度格式与单精度相似，双精度浮点数可由 $F_{D}=(-1)^{S}1.M\times 2^{E-1023}$。

表 8.2　IEEE 单精度浮点表示中的特殊数字

E	M	计数	说明
$0<E<255$	任意	$(-1)^{S}1.M\times 2^{E-127}$	一般值
255	非零	NaN	未定义的数
255	零	无穷	最大值
0	零	零	

设计者在数字滤波器实现中如何选择数值系统，需要权衡多种因素，包括运算速度和成本。一般来说，高速低成本的系统可以采用定点处理器，也就是说，由于仅可表示小范围的数字，此时滤波器需要特殊设计以保证不受溢出的影响。使用浮点型表示方法虽然可以屏蔽某些溢出问题，但是会使得成本更高且计算更加复杂。

设计中需要考虑的另一个因素是实际应用中的溢出特征，对于某些系统，溢出特征(饱和或者二进制补码)可能导致不利的非线性效果，使得系统出现振荡。

8.3　实现结构

一个因果 DSP 系统可由 z 域传输函数表示，对应时域为差分方程。差分方程的实现，仅需要三种固定的运算：常数乘法、加法以及时间延迟，因此可以利用这三种运算表示传输函数的实现结构。结构操作等同差分方程的操作，结构以框图显示，可以展示变量间的关系，具有视觉上的优势。结构在形式上与信号流图近似，因此某些作者也使用信号流图来表示。例 8-1 中的结构框图将使用三种符号。

例 8-1　传输函数的方框图表示

描述某离散系统的传输函数为

$$H(z)=\frac{z+1}{z^{2}-0.9}$$

利用常数乘法、单位延迟和加法模块画出对应的方框图表示。

解： 由离散系统的传输函数，可以获得差分方程的 z 变换形式 $y(z)=z^{-1}x(z)+z^{-2}x(z)+0.9z^{-2}y(z)$。一种形式的方框图表示如图 8.5 所示。

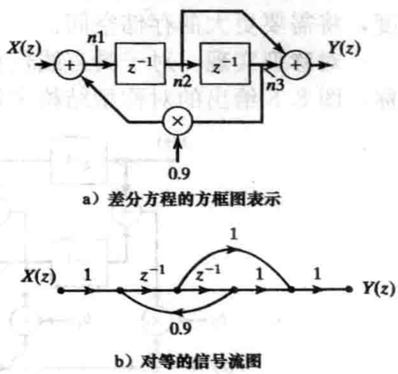


图 8.5　例 8-1 图

为验证这一点，可以写出如下 4 个等式：

$$n1=X(z)+0.9n3$$

$$n2 = z^{-1} n1$$
$$n3 = z^{-1} n2$$
$$Y(z) = n2 + n3$$

从这四个等式，去除中间节点 $n1$、$n2$ 和 $n3$，可以得出 z 域输入与输出之间的关系

$$Y(z) = z^{-1}X(z) + z^{-2}X(z) + 0.9z^{-2}Y(z)$$
$$Y(z) = \frac{z^{-1} + z^{-2}}{1 - 0.9z^{-2}}X(z) = \frac{z+1}{z^2 - 0.9}X(z)$$ ◀

8.3.1　FIR 结构

FIR 滤波器的差分方程可以改写为多种不同的实现形式，这里主要分析直接型、转置型、对称型、级联型、并联型和格型 6 种类型的实现结构。每一种实现结构具有各自的优缺点，如何选择合适的实现结构类型是滤波器设计过程中的一部分。

直接型实现　图 8.6 所示带有抽头的延迟线构成的 FIR 滤波器，我们称这种结构为 FIR 滤波器的直接型实现。

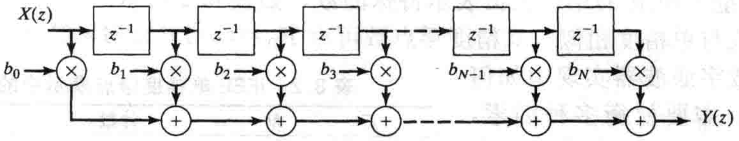


图 8.6　N 阶 FIR 滤波器的直接型实现，脉冲响应为 $h[n]=\{b_0, b_1, \cdots, b_N\}$

转置型实现　直接型结构的另一种形式可由流图转置定理生成。依据该定理，将所有分支逆转，输入、输出交换即可获得转置型。对于一个单输入单输出的 LTI 系统，转置系统具有与原始系统相同的传输函数，图 8.7 给出了图 8.6 中直接型 FIR 滤波器对应的转置型结构。

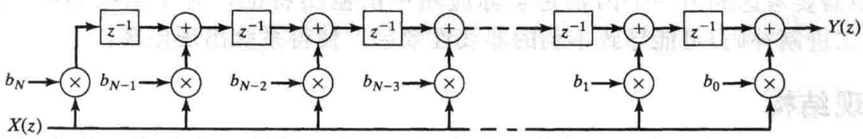


图 8.7　N 阶 FIR 滤波器的转置型实现

直接型和转置型需要相同数量的乘法器、加法器和存储位置。转置型存在较小的缺点，在于其通过的延迟线平移的数值是一个乘积。如果采用保护位以确保乘积运算的精度，将需要更大的存储空间。

对称型实现　对于具有线性相位的 FIR 数字滤波器，滤波器的系数要么对称要么反对称。图 8.8 给出的对称型结构采用了系数对称的特点，降低了乘法器的数量。

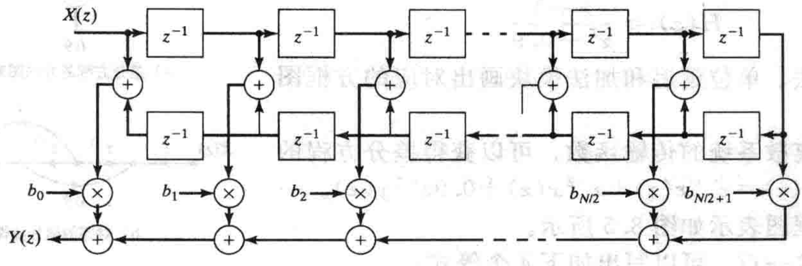


图 8.8　FIR 滤波器的对称结构，其中 $b_0=b_N$，$b_1=b_{N-1}$，…

图 8.8 中，滤波器的阶数为奇数阶，长度 N 为偶数，所以差分方程可以写为

$$y[n]=b_0(x[n]+x[n-N-1])+b_1(x[n-1]+x[n-N])+\cdots+b_{N/2}(x[n-N/2]+x[n-N/2+2])+b_{N/2+1}x[n-N/2+1]$$

对称结构具有明显的优点，可以有效地减少乘积运算，乘积运算比加法运算常需要占用更多的处理时间，因此该结构可以明显节省运算时间。

级联型实现　将 FIR 滤波器传输函数的分子部分分解成一系列二次项，每个二次项对应一个二阶子系统，级联这些二阶子系统，其实现结构可以由图 8.9 表示。

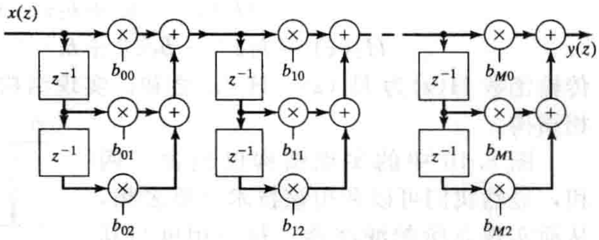


图 8.9　级联型 FIR 滤波器实现结构，共有 $M+1$ 个二阶系统

例 8-2　设计一个 9 阶的 FIR 滤波器

利用 MATLAB 中的 firl 函数生成一个 9 阶的 FIR 滤波器，窗函数为汉明窗，抽样频率为 22050Hz，截止频率为 3000Hz，求该滤波器级联结构实现的系数。

解：下面的 MATLAB 命令生成滤波器的系统，并将分子系数存储在 num 向量中。

```
fs = 22050;
fc = 3000;
N = 9;
num = fir1(N, fc/(fs/2), hamming(N+1))
```

为了获得各个二阶子系统的系数，可以利用 MATLAB 中的 tf2sos 函数求取 num 的根，以如下命令实现。

```
sos = tf2sos(num, 1);
[r c] = size(sos);
for i = 1:r
 for j = 1:c
 fprintf(1, '%2.6f', sos(i, j));
 end
 fprintf(1, '\n');
end
```

以上命令输出如下结果：

−0.003974 0.021105−0.000000 1.000000 0.000000 0.000000
1.000000 0.811702−0.188298 1.000000 0.000000 0.000000
1.000000 1.677041 1.986063 1.000000 0.000000 0.000000
1.000000 1.284306 1.000000 1.000000 0.000000 0.000000
1.000000 0.844405 0.503509 1.000000 0.000000 0.000000

即为 sos 矩阵，给出了二阶多项式的系数，矩阵的左半部分列出了三个分子系数，右半部分为三个分母系数，与图 8.9 对应的二阶节如下所示。

$$\text{Section1}=\frac{-0.003974z+0.021105}{z}\qquad b_{00}=-0.003974, b_{01}=0.21105, b_{02}=0$$

$$\text{Section2}=\frac{z^2+0.811702z-0.188298}{z^2}\qquad b_{10}=1, b_{11}=0.811702, b_{12}=-0.188298$$

……

$$\text{Section5}=\frac{z^2+0.844405z-0.503509}{z^2}\qquad b_{40}=1, b_{41}=0.844405, b_{42}=0.503509$$

◀

将 FIR 滤波器分解为二阶子系统在计算上可以模块化实现。例 8-2 中，可以利用 5 个

微控制器分别单独计算各二阶子系统以实现整个滤波器。模块化可以任意调整前后顺序，而不改变整个传输函数，这样具有一个优点就是可以将低增益的模块放置于高增益的模块之前，以有助于缩放增益。

并联型实现　对于一个6阶FIR滤波器传输函数

$$H(z) = b_0 + b_1 z^{-1} + b_2 z^{-2} + b_3 z^{-3} + b_4 z^{-4} + b_5 z^{-5} + b_6 z^{-6}$$

将该传输函数分成两个表示式，分别包含其中的奇数项和偶数项：

$$H_1(z) = b_0 + b_2 z^{-2} + b_4 z^{-4} + b_6 z^{-6}$$

$$H_2(z) = b_1 z^{-1} + b_3 z^{-3} + b_5 z^{-5} = z^{-1}(b_1 + b_3 z^{-2} + b_5 z^{-4})$$

传输函数H(z)为$H_1(z)$、$H_2(z)$之和，实现结构如图8.10所示。该并联结构通常称为多相结构。

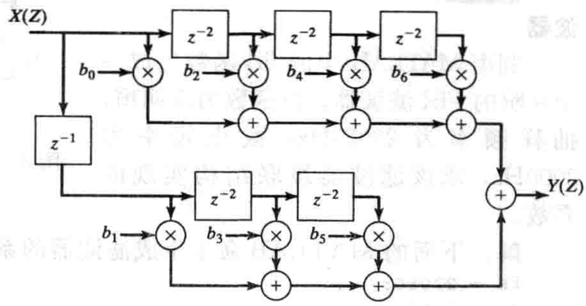


图8.10　并联或者多相实现的6阶FIR滤波器

图8.10中的实现结构仅包含了两相，显然我们可以利用该技术至更多相，从而实现高阶的滤波器。每一相可与其他相平行实现，使得该结构可以利用多个微控制器实现，每个微控制器在每个周期中仅须处理少量的乘积运算。

格型实现　格型结构可以用来实现FIR滤波器和IIR滤波器。单级FIR子系统如图8.11所示。该结构中不直接使用传输函数的系数，其中的系数这里称为m_i，可由传输函数系数推导得出。

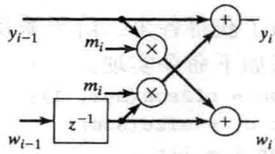


图8.11　FIR滤波器的单级格型结构

对于图8.11，可以写出如下式子：

$$y_i = y_{i-1} + m_i z^{-1} w_{i-1} \tag{8.2}$$

$$w_i = m_i y_{i-1} + z^{-1} w_{i-1} \tag{8.3}$$

N阶FIR滤波器的格型结构如图8.12所示，输入连接至第一级的输入y_0和w_0。由式(8.2)和式(8.3)，可以得到该FIR格型滤波器的传输函数。

$$y_1/x = 1 + m_1 z^{-1}$$

$$w_1/x = m_1 + z^{-1}$$

$$y_2/x = 1 + (m_1 + m_1 m_2) z^{-1} + m_2 z^{-2}$$

$$w_2/x = m_2 + (m_1 + m_1 m_2) z^{-1} + 1$$

$$y_3/x = 1 + (m_1 + m_1 m_2 + m_2 m_3) z^{-1} + (m_3 m_2 m_1 + m_3 m_1 + m_2) z^{-2} + m_3 z^{-3}$$

$$w_3/x = m_3 + (m_3 m_2 m_1 + m_3 m_1 + m_2) z^{-1} + (m_1 + m_1 m_2 + m_2 m_3) z^{-2} + z^{-3}$$

…

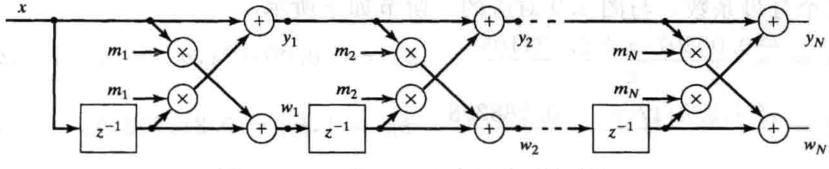


图8.12　N阶FIR滤波器的型格结构

通过式(8.2)和式(8.3)以及上面的推导公式，可以得出：

1. (y_i/x)和(w_i/x)具有相同的系数，只是顺序相反。
2. w_i的z^0系数总是m_i。

求解式(8.3)得出 w_{i-1} 并代入式(8.2)，结果为

$$y_{i-1} = \frac{y_i - m_i w_i}{1 - m_i^2} \tag{8.4}$$

若已知传输函数 y_M/x，通过式(8.4)可以迭代求出系数 m_i。MATLAB 中的 tf2latc 函数利用该流程实现滤波器格型结构，如例 8-3 所示。

例 8-3 通过传输函数绘制格型结构滤波器

画出对应如下三阶 FIR 滤波器传输函数的格型结构

$$H(z) = \frac{z^3 + 2.218z^2 + 1.996z + 0.779}{z^3}$$

解：下面的 MATLAB 命令可求解出 m_i：

```
num = [1 2.218 1.996 .729];
m = tf2latc(num);
disp(m);
```

命令窗口输出 m_i 的值

0.900050375337556

0.809029385840417

0.729000000000000

相比直接型结构，FIR 滤波器的格型结构没有特殊的优点，但是对于 IIR 滤波器，格型结构在某些应用中确实具有较好的性能。

8.3.2 IIR 结构

IIR 滤波器存在可能的不稳定性，由于其存在反馈回路使系统中误差得到反馈，其对系数量化误差和舍入误差更为敏感。在 IIR 滤波器实现中关注的不仅是计算效率，还关注实现结构对误差的响应。本节将涉及 IIR 滤波器的直接型、转置型、并联型、级联型、格型，以及状态变量等实现。

直接型实现 IIR 数字滤波器的传输函数一般可表示为

$$y(z) + a_1 z^{-1} y(z) + \cdots + a_N z^{-N} y(z) = b_0 x(z) + b_1 z^{-1} x(z) + \cdots + b_M z^{-M} x(z)$$

公式的两侧类似于 FIR 滤波器的形式，可以分别实现，共同生成输出，如图 8.13 所示。

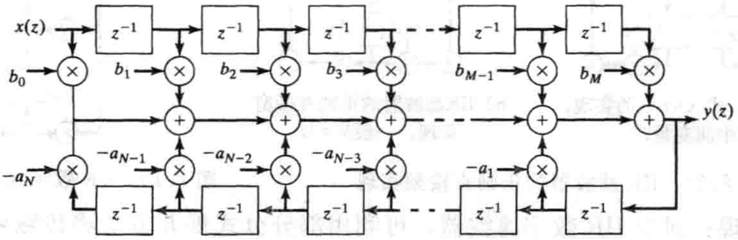


图 8.13 IIR 滤波器的直接 1 型实现，该系统为非正则系统，一般来说 $N \geqslant M$

在图 8.13 中，分子的阶数为 M，分母(反馈项)的阶数为 N。对于因果系统，需要满足 $N \geqslant M$。分子中有 M 个单位延迟，分母为 N 个单位延迟，共需 $N+M$ 个。一个系统称为正则的(canonic)，指系统实现所需的单位延迟的数量等于系统的阶数。因此，上述系统为非正则系统(上述系统阶数为 N)。可以通过寻找某种途径实现单位延迟的共享，来实现系统正则以降低系统计算的复杂度。

对于 IIR 数字滤波器的差分方程

$$y[n] = \sum_{i=0}^{M} b_i x[n-i] - \sum_{i=1}^{N} a_i y[n-i]$$

由 z 变换的平移特性，有

$$Y(z)=X(z)\sum_{i=0}^{M}b_i z^{-i}-Y(z)\sum_{i=1}^{N}a_i z^{-i}$$

式子两边同除 b_i 项之和

$$\frac{Y(z)}{\sum_{i=0}^{M}b_i z^{-i}}=X(z)-\frac{Y(z)\sum_{i=0}^{N}a_i z^{-i}}{\sum_{i=0}^{M}b_i z^{-i}}$$

定义中间变量 $T(z)$

$$T(z)=\frac{Y(z)}{\sum_{i=0}^{M}b_i z^{-i}}$$

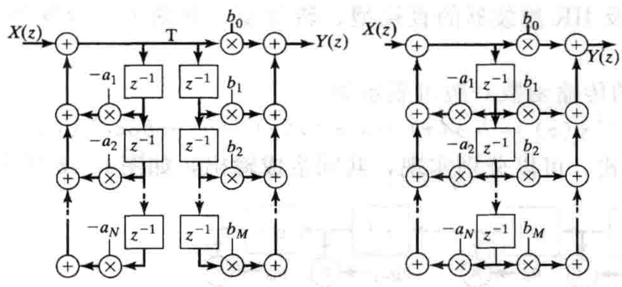
可以得到以下两个等式

$$Y(z)=T(z)\sum_{i=0}^{M}b_i z^{-i} \tag{8.5}$$

$$T(z)=X(z)-T(z)\sum_{i=0}^{N}a_i z^{-i} \tag{8.6}$$

图 8.14a 给出了以上两个公式的实现。图 a 中存在单位延迟的重复，通过单位延迟的复用可以转换成 b 的形式，即 IIR 数字滤波器的正则直接型实现。

转置型实现：图 8.15 给出了 IIR 数字滤波器的转置型结构，可由正则直接型转置得到。该实现形式对比直接型在计算上不存在优势，类似 FIR 滤波器实现形式，单位延迟的输入是乘积项而非输入变量。



a）式（8.5）、式（8.6）的实现，包含中间变量T

b）IIR滤波器的正则直接型实现，一般$N\geqslant M$

图 8.14　IIR 滤波器的正则直接型实现

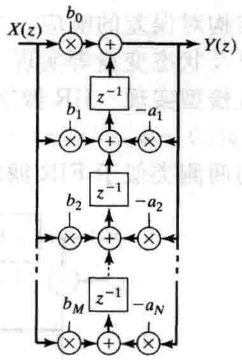


图 8.15　IIR 数字滤波器的转置型结构

并联型实现：对于 IIR 数字滤波器，可利用部分分式展开方法将传输函数表示为各项之和的形式从而获得并联型实现。如果 IIR 数字滤波器在原点处存在极点，需要去除极点并以延迟形式单独对待。假设分母不存在重根，也是经常出现的情况。对于 z 域传输函数表示式，其分子阶数为 M，分母阶数为 N，$N\geqslant M$，并且不存在重极点，部分分式展开可写为

$$H(z)=\frac{N(z)}{D(z)}=K+\sum_{i=1}^{N}\frac{A_i}{z-p_i}$$

式中，p_i 为分母多项式的根，A_i 和 K 为常数。

当且仅当 $M=N$ 时，K 为非零值，且可由分子和分母直接相除得到。A_i 的值为

$$A_i=(z-p_i)H(z)\big|_{z=p_i}$$

如果 p_i 为复数，则 A_i 也为复数。该复数极点项与其共轭复数极点项将共同组合成实系数

的二阶子系统。如果 $A_i=R_A+jI_A$，$p_i=\alpha+j\beta$，那么对应的实系数二阶子系统为

$$\frac{A_i}{z-p_i}+\frac{A_i^*}{z-p_i^*}=\frac{2(R_A z+\alpha R_A+I_A\beta)}{z^2-2\alpha z+\alpha^2+\beta^2} \tag{8.7}$$

例 8-4　切比雪夫滤波器的并联型实现

4 阶切比雪夫低通数字滤波器如下

$$H(z)=2.3972\times10^{-4}\frac{z^4+4z^3+6z^2+4z+1}{z^4-3.4256z^3+4.49158z^2-2.6639z+0.60205}$$

试利用部分分式展开法求解滤波器的并联型实现。

解：MATLAB 中的 residue 函数可用于求解部分分式展开项。

```
num = 0.00023972*[1 4 6 4 1];
den = [1 -3.4256 4.49158 -2.6639 0.60205];
[A P K] = residue(num, den);
```

求得 K 值为

$$K=2.39717e-004$$

A_i 系数及极值见表 8.3 所示。

对于复数极点项，根据式(8.7)，可以求出对应的实系数二阶子系统，该数字滤波器的并联型实现如图 8.16 所示。

表 8.3　4 阶切比雪夫滤波器部分分式展开的 A_i 和 p_i 值

i	A_i	P_i
1	−0.066690+j0.037434	0.885166+j0.283515
2	−0.066690−j0.037434	0.885166−j0.283515
3	0.067580−j0.156334	0.827749+j0.108329
4	0.067580+j0.156334	0.827749−j0.108329

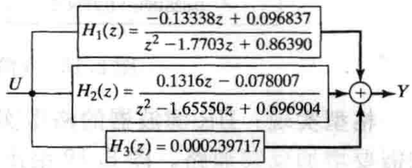


图 8.16　4 阶 Chebyshev 数字滤波器的并联型实现

并联结构实现的优势在于每个二阶子系统均可由单独的控制器独立计算而实现，量化误差得到降低，这部分内容将在下一节说明。但是，对于并联型结构，分子和分母组合不能互换。也就是说，如果交换子系统 1 与子系统 2 的分子而不交换分母，将得到一个不同的数字滤波器。我们将会发现，级联型结构确实可以任意交换子系统的分子。

我们也注意到部分分式展开可以用另一种略微不同的形式表示：

$$H(z)=\frac{N(z)}{D(z)}=K+\sum_{i=1}^{N}\frac{A_i z}{z-p_i}$$

这种形式利用 MATLAB 中的 residuez 函数实现。

级联型实现：将传输函数的分子和分母因式分解为二级子系统，通过级联这些二阶子系统即可得到 IIR 数字滤波器的级联型实现。系统输入 X 是第一个子系统的输入，后面的每一子系统都从前一个子系统的输出得到其输入，如图 8.17 所示。

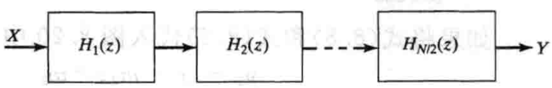


图 8.17　级联的二阶子系统

为了获得每个二阶子系统的传输函数，我们将分子和分母因式分解为二次项。如果为奇数阶，将会出现一个一阶子系统。级联的子系统可以任意调换分子二次项的位置，级联型实现在这方面优于并联型实现，因为这可以有效地组合极点和零点以更有利于控制溢出。

例 8-5　6 阶数字滤波器的级联型实现

试将以下 6 阶椭圆低通数字滤波器以二阶子系统级联型实现。

$$H(z)=\frac{0.04218z^6+0.03671z^5+0.09747z^4+0.746z^3+0.09747z^2+0.03571z+0.04218}{z^6-2.32027z^5+3.5786z^4-3.25141z^3+2.04355z^2-0.75926z+0.14353}$$

解：我们可以求解分子和分母多项式的根，匹配共轭对来形成各二阶子系统。MATLAB中tf2sos的函数可以实现上述计算。以下是MATLAB代码：

```
num = [0.04218 0.03671 0.09747 0.07496 0.09747 0.03671 0.04218];
den = [1.00000 -2.32027 3.57586 -3.25141 2.04355 -0.75926 0.14353];
sos = tf2sos(num, den);
```

矩阵sos将有3行6列。每一行表示一个二阶子系统。前面三列是分子系数，后三列为分母系数。sos的内容如表8.4所示。

表8.4　传输函数转换为二阶子系统后的sos内容

子系统	b_0	b_1	b_2	a_0	a_1	a_2
1	0.042184	0.058486	0.042184	1.000000	−0.828188	0.253422
2	1.000000	−0.057559	1.000000	1.000000	−0.758492	0.627054
3	1.000000	−0.458728	1.000000	1.000000	−0.733589	0.903236

默认情况下，MATLAB将距离单位圆最近的共轭极点和共轭零点进行匹配。重复该过程直到所有的共轭对匹配完成。比例系数(scaling)一般放在第一节。比例系数和子系统顺序都可以通过功能选项来改变。图8.18展示了二阶级联子系统的完整示意图。

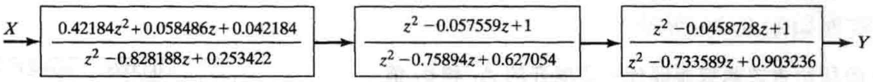


图8.18　6阶椭圆数字滤波器的级联二阶子系统

格型实现：IIR滤波器的格型实现与FIR滤波器相似，构建IIR数字滤波器的格型结构需要增加反馈通路。图8.19给出了一个典型的带有反馈的格型部分，由图8.19可得

$$y_i = y_{i+1} - z^{-1}m_i w_i \tag{8.8}$$

$$w_{i+1} = m_i y_i + z^{-1} w_i \tag{8.9}$$

此种IIR数字滤波器格型结构的所有零点都在原点位置，因此通常称为全极点格型滤波器。全极点滤波器的结构与FIR滤波器相似，且系数m_i与FIR的系数相同。图8.20给出了3阶全极点格型结构。

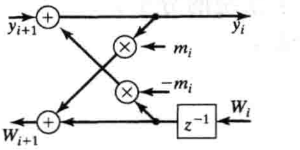


图8.19　带有反馈的格型结构，可组成全极点IIR数字滤波器

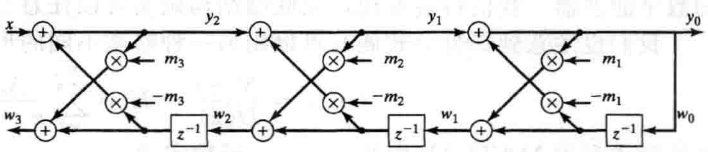


图8.20　3阶全极点格型结构

如果将式(8.8)和式(8.9)代入图8.20中的结构，可以得到如下6个等式：

$$y_2 = x - m_2 z^{-1} w_2 \qquad w_1 = m_1 y_0 + z^{-1} w_0$$

$$y_1 = y_2 - m_2 z^{-1} w_1 \qquad w_2 = m_2 y_1 + z^{-1} w_1$$

$$y_0 = y_1 - m_1 z^{-1} w_0 \qquad w_3 = m_3 y_2 + z^{-1} w_2$$

推导以上6个等式，可以求解出w_i/w_0和y_i/x。由于y_0和w_0有相同的信号，因此可求出w_i/u。由此得出以下关系：

$$\frac{w_0}{x} = \frac{y_0}{x} = \frac{z^3}{D(z)}$$

$$\frac{w_1}{x} = \frac{z^3 m_1 + z^2}{D(z)}, \quad \frac{y_1}{x} = \frac{z^3 + z^2 m_1}{D(z)}$$

$$\frac{w_2}{x} = \frac{z^3 m_2 + z^2(m_1 m_2 + m_1) + z}{D(z)}, \quad \frac{y_2}{x} = \frac{z^3 + z^2(m_1 m_2 + m_1) + z m_2}{D(z)}$$

$$\frac{w_3}{x} = \frac{z^3 m_3 + z^2(m_1 m_2 m_3 + m_1 m_3 + m_2 m_3) + z(m_1 + m_1 m_2) + 1}{D(z)}$$

其中，$D(z)=z^3+z^2(m_1m_2+m_1)+z(m_1m_2m_3+m_1m_3+m_2m_3)+m_3$。

由于 $D(z)$为期望的格型滤波器的分母，因此利用 $D(z)$的系数可以写出 w_i/x 项。

如果 $D(z)=a_0z^3+a_1z^2+a_2z+a_3$，那么

$$w_0/x = a_0 z^3/D(z)$$
$$w_1/x = (a_0 z^3 + z^2)/D(z)$$
$$w_2/x = (a_3 z^3 + a_1 z^2 + z)/D(z)$$
$$w_3/x = (a_3 z^3 + a_2 z^2 + a_1 z + a_0)/D(z)$$

从以上公式，可以得出以下结论：

1. w_i/x 和 y_i/x 具有相同的分母，$D(z)$。
2. w_i/x 和 y_i/x 的分子相同，但系数反转。
3. w_N/x 的分子与 $D(z)$相同，但系数反转。

同时注意，w_N/w_0 为 N 阶 FIR 格型滤波器的传输函数。因此，若需要将全极点传输函数转换为格型结构，我们可以先求传输函数的倒数(即将所有极点当作零点)，然后当作 FIR 格型滤波器以计算出对应的系数。MATLAB 中的 tf2latc 函数用于计算 IIR 全极点格型滤波器的系数。下面的 MATLAB 命令产生了 IIR 数字滤波器的全极点格型系数，其与例 8-3 中系统函数互为倒数。

```
den = [1 2.218 1.996 .729];
K = tf2latc(1, den);
```

在全极点格型结构中，通过在每个子系统上添加一个环路即可添加零点，如图 8.21 所示。

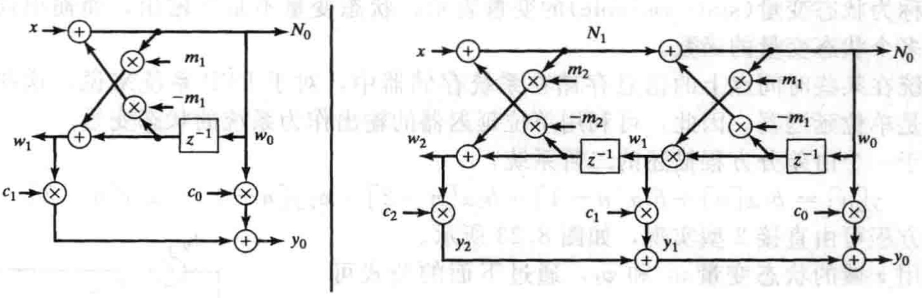


图 8.21 带有零、极点的 IIR 格型实现子系统

对于图 8.21，数字滤波器的传输函数可以写为

$$H(z) = c_0 w_0 + c_1 w_1 + \cdots + c_N w_N \tag{8.10}$$

通过对于全极点格型滤波器的分析，w_i 具有相同的分母多项式，其也为数字滤波器 $H(z)$的分母。$H(z)$的分子由 $c_i \times w_i$ 之积的和而构成。c_i 的推导将不在这里详述[1]。MATLAB 中的 tf2latc 函数可用于 c_i 和 m_i 系数的求解，如例 8-6 所示。

例 8-6 转换椭圆数字滤波器为格型结构

下面为三阶低通椭圆滤波器的传输函数，请利用 MATLAB 将其转化为格型结构。

$$H(z) = \frac{0.1129z^3 + 0.2191z^2 + 0.2191z + 0.1129}{z^3 - 0.9015z^2 + 0.7608z - 0.1955} = \frac{N(z)}{D(z)}$$

解：

```
num = [0.1129 0.2191 0.2191 0.1129];
den = [1 -0.9015 0.7608 -0.1955];
[m c] = tf2latc(num, den);
```

给出如下格型系数：

格型系数 m

m1 = −0.4868, m2 = 0.6078, m3 = −0.1955

格型系数 c

c1 = 0.1270, c2 = 0.3843, c3 = 0.3208, c4 = 0.1129

格型结构的数字滤波器如图 8.22 所示。

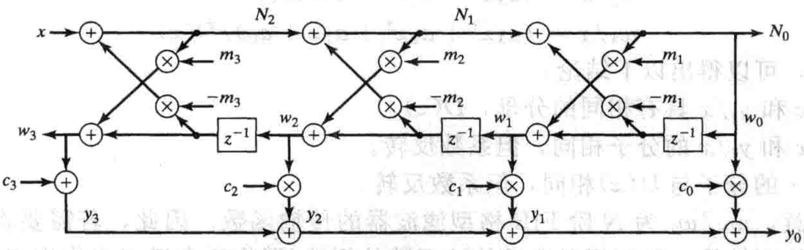


图 8.22　3 阶椭圆格型滤波器

8.3.3 状态空间表示

许多实现结构表明系统差分方程可以重新组织，以便于系统实施。一般来说，对于给定的实现我们期望获得某些计算上的优势，如速度、空间或者降低误差。以上各种实现都可以利用状态空间系统(state space system)表示。根据状态空间的观点，一个系统可以由表征系统特征的独立状态集合(亦称状态空间)来描述。我们定义系统的状态为所需的基本信息，这些信息与系统输入一起，可用来确定特定时刻后的任意点的输出。状态通常通过一系列称为状态变量(state variable)的变量表示，状态变量不是指输出，而输出通常是一个或者多个状态变量的函数。

系统在某些时间点上的信息存储在系统存储器中，对于 DSP 系统来说，这些存储器元件就是单位延迟器。因此，可利用单位延迟器的输出作为系统的状态变量。

对于一个由差分方程描述的二阶系统，

$$y[n]=b_0x[n]+b_1x[n-1]+b_2x[n-2]-a_1y[n-1]-a_2y[n-2]$$

该差分方程可由直接 2 型实现，如图 8.23 所示。

利用 z 域的状态变量 v_1 和 v_2，通过下面的公式可表征该系统

$$zv_1=v_2 \tag{8.11}$$

$$zv_2=-a_2v_1-a_1v_2+x \tag{8.12}$$

$$y=(b_2-b_0a_2)v_1+(b_1-b_0a_1)v_2+b_0x \tag{8.13}$$

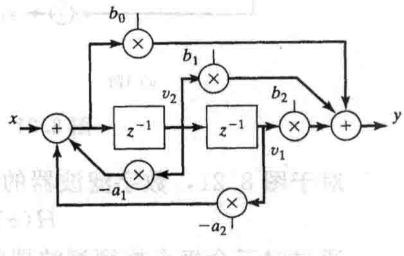


图 8.23　2 阶直接 2 型，具有状态变量 v_1 和 v_2

式(8.11)和式(8.12)称为系统的状态方程，式(8.13)称为系统的输出方程。一般地，对于 N 阶单输入单输出系统，将有 N 个状态方程和 1 个输出方程。需要指出的是，这些方程与差分方程表示了相同的信息，因为通过求解其中一个方程的 v_1 代入其他两个方程，再通过其余两个方程之一求解 v_2 代入另一个方程中，即可得到原始的差分方程。

利用矩阵形式表示，状态方程和输出方程可写为

$$z\boldsymbol{v}=\boldsymbol{A}\boldsymbol{v}+\boldsymbol{B}x \tag{8.14}$$

$$y=\boldsymbol{C}\boldsymbol{v}+\boldsymbol{D}x \tag{8.15}$$

式中，$\boldsymbol{v}$ 为状态变量向量

$$\boldsymbol{v}=\begin{bmatrix}v_1\\v_2\end{bmatrix}$$

和

$$\boldsymbol{A}=\begin{bmatrix}0 & 1\\-a_2 & -a_1\end{bmatrix},\quad \boldsymbol{B}=\begin{bmatrix}0\\1\end{bmatrix},\quad \boldsymbol{C}=[(b_2-b_0a_2)\quad(b_1-b_0a_1)],\quad \boldsymbol{D}=[1] \tag{8.16}$$

对于 $\boldsymbol{N}$ 阶单输入单输出系统，可有 $\boldsymbol{N}$ 个状态方程和一个输出方程，矩阵 $\boldsymbol{A}$ 为 $\boldsymbol{N}\times\boldsymbol{N}$，$\boldsymbol{B}$ 为 $\boldsymbol{N}\times 1$，$\boldsymbol{C}$ 为 $1\times\boldsymbol{N}$，$\boldsymbol{D}$ 为 1×1。

为了得到矩阵形式的传输函数，求解式(8.14)中的 $\boldsymbol{v}$。

$$\boldsymbol{v}=(z\boldsymbol{I}-\boldsymbol{A})^{-1}\boldsymbol{B}x$$

代入式(8.15)可得

$$y=[\boldsymbol{C}(z\boldsymbol{I}-\boldsymbol{A})^{-1}\boldsymbol{B}+\boldsymbol{D}]x$$

传输函数定义为

$$H(z)=[\boldsymbol{C}(z\boldsymbol{I}-\boldsymbol{A})^{-1}\boldsymbol{B}+\boldsymbol{D}] \tag{8.17}$$

例 8-7 通过状态空间表示求解系统传输函数

利用式(8.17)求解数字滤波器的传输函数，其状态空间表示为

$$\boldsymbol{A}=\begin{bmatrix}0 & 1\\-0.9 & 1.2\end{bmatrix},\quad \boldsymbol{B}=\begin{bmatrix}0\\1\end{bmatrix},\quad \boldsymbol{C}=[(0.1)\quad(3.2)],\quad \boldsymbol{D}=[1] \tag{8.18}$$

解：

$$H(z)=[0.1\quad 3.2]\begin{bmatrix}z & -1\\0.9 & z-1.2\end{bmatrix}^{-1}\begin{bmatrix}0\\1\end{bmatrix}+[1]$$

$$H(z)=[0.1\quad 3.2]\frac{\begin{bmatrix}z-1.2 & 1\\-0.9 & z\end{bmatrix}}{|z\boldsymbol{I}-\boldsymbol{A}|}\begin{bmatrix}0\\1\end{bmatrix}+[1]$$

$$H(z)=\frac{z^2+2z+1}{z^2-1.2z+0.9}$$

需要指出分母 $|z\boldsymbol{I}-\boldsymbol{A}|=z^2-1.2z+0.9$，方程 $|z\boldsymbol{I}-\boldsymbol{A}|=0$ 称为特征方程，特征方程的根决定了系统的稳定性和瞬态响应。 ◀

假如我们将矩阵形式的传输函数进行转置，可以得到转置系统的传输函数。矩阵的转置利用T的上标表示，如 $\boldsymbol{A}^{\mathrm{T}}$，三个相乘的矩阵的转置具有如下性质

$$(\boldsymbol{XYZ})^{\mathrm{T}}=(\boldsymbol{Z}^{\mathrm{T}}\boldsymbol{Y}^{\mathrm{T}}\boldsymbol{X}^{\mathrm{T}})$$

可以得到

$$\mathrm{Transpose}(H(z))=[\boldsymbol{B}^{\mathrm{T}}(z\boldsymbol{I}-\boldsymbol{A}^{\mathrm{T}})^{-1}\boldsymbol{C}^{\mathrm{T}}+\boldsymbol{D}^{\mathrm{T}}]$$

对比式(8.17)，可以看出通过转置所有矩阵、交换转置 B 矩阵和转置 C 矩阵可以表示转置系统。转置系统的矩阵方程为

$$\boldsymbol{v}=\begin{bmatrix}v_1\\v_2\end{bmatrix}$$

和

$$\boldsymbol{A}=\begin{bmatrix}0 & -a_2\\1 & -a_1\end{bmatrix},\quad \boldsymbol{B}=\begin{bmatrix}b_2-b_0a_2\\b_2-b_0a_1\end{bmatrix},\quad \boldsymbol{C}=[0\quad 1],\quad \boldsymbol{D}=[1] \tag{8.19}$$

不难验证，通过给定图 8.15 中转置系统的状态变量就可以求出系统的状态变量方程。转置系统公式的数学推导凸显了状态空间表示的一个优点：即对应相同的系统可以生成多种不同的表示方法，其中一些具有很好的计算优点，或者较小的数值误差。

通过状态变量的线性变换，可以改变式(8.14)和式(8.15)所示状态方程中矩阵 A、B、C、D。为此目的，定义新的时域状态变量向量 $\boldsymbol{q}$，其对应的 z 域状态变量向量为 $\boldsymbol{Q}$，使得

$$\boldsymbol{Q}=\boldsymbol{T}^{-1}\boldsymbol{v}$$

式中，矩阵 $\boldsymbol{T}$ 为 $\boldsymbol{N}\times\boldsymbol{N}$ 的变换矩阵。

式(8.14)和式(8.15)变为

$$z\boldsymbol{TQ}=\boldsymbol{ATQ}+\boldsymbol{B}x \tag{8.20}$$

$$y=\boldsymbol{CTQ}+\boldsymbol{D}x \tag{8.21}$$

式(8.20)两边同乘 $\boldsymbol{T}^{-1}$

$$z\boldsymbol{Q}=\boldsymbol{T}^{-1}\boldsymbol{ATQ}+\boldsymbol{T}^{-1}\boldsymbol{B}x \tag{8.22}$$

$$y=\boldsymbol{CTQ}+\boldsymbol{D}x \tag{8.23}$$

将式(8.22)、式(8.23)与式(8.14)、式(8.15)对比可以看出，如果定义 $\boldsymbol{N}\times\boldsymbol{N}$ 的非奇异矩阵 $\boldsymbol{T}$，则可以实现坐标变换，并做如下矩阵替换：

$$\boldsymbol{A}\leftarrow\boldsymbol{T}^{-1}\boldsymbol{AT} \tag{8.24}$$

$$\boldsymbol{B}\leftarrow\boldsymbol{T}^{-1}\boldsymbol{B} \tag{8.25}$$

$$\boldsymbol{C}=\boldsymbol{CT} \tag{8.26}$$

$$\boldsymbol{D}\leftarrow\boldsymbol{D} \tag{8.27}$$

◀

例 8-8 求解变换的状态矩阵

对于例 8-7 中状态矩阵描述的系统，利用变换矩阵求解变换后的状态矩阵

$$\boldsymbol{T}=\begin{bmatrix}-1.3608 & 0\\ -0.8165 & 1\end{bmatrix}$$

证明，坐标变换一般不会改变传输函数。

解： 原始的矩阵 $\boldsymbol{A}$、$\boldsymbol{B}$、$\boldsymbol{C}$ 和 $\boldsymbol{D}$ 为

$$\boldsymbol{A}=\begin{bmatrix}0 & 1\\ -0.9 & 1.2\end{bmatrix},\quad \boldsymbol{B}=\begin{bmatrix}0\\ 1\end{bmatrix},\quad \boldsymbol{C}=[0.1\quad 3.2],\quad \boldsymbol{D}=[1]$$

利用式(8.24)至式(8.27)，可以得到

$$\boldsymbol{A}\leftarrow\boldsymbol{T}^{-1}\boldsymbol{AT}=\begin{bmatrix}-1.3608 & 0\\ -0.8165 & 1\end{bmatrix}^{-1}\begin{bmatrix}0 & 1\\ -0.9 & 1.2\end{bmatrix}\begin{bmatrix}-1.3608 & 0\\ -0.8165 & 1\end{bmatrix}=\begin{bmatrix}0.6 & -0.7348\\ 0.7348 & 0.6\end{bmatrix}$$

$$\boldsymbol{B}\leftarrow\boldsymbol{T}^{-1}\boldsymbol{B}=\begin{bmatrix}-1.3608 & 0\\ -0.8165 & 1\end{bmatrix}\begin{bmatrix}0\\ 1\end{bmatrix}=\begin{bmatrix}0\\ 1\end{bmatrix}$$

$$\boldsymbol{C}\leftarrow\boldsymbol{CT}=[0.1\quad 3.2]\begin{bmatrix}-1.3608 & 0\\ -0.8165 & 1\end{bmatrix}=[-2.7489\quad 3.2]$$

$$\boldsymbol{D}\leftarrow\boldsymbol{D}=[1]$$

为证明传输函数一般保持不变，根据式(8.17)可得

$$H(z)=[\boldsymbol{C}(z\boldsymbol{I}-\boldsymbol{A})^{-1}\boldsymbol{B}+\boldsymbol{D}]$$

将矩阵的变换形式代入该方程

$$H_{\mathrm{T}}(z)=[\boldsymbol{C}(z\boldsymbol{I}-\boldsymbol{T}^{-1}\boldsymbol{AT})^{-1}\boldsymbol{T}^{-1}\boldsymbol{B}+\boldsymbol{D}]$$

这个方程可以写为

$$H_{\mathrm{T}}(z)=[\boldsymbol{CTT}^{-1}(z\boldsymbol{I}-\boldsymbol{A})^{-1}\boldsymbol{TT}^{-1}\boldsymbol{B}+\boldsymbol{D}]$$

由于 $\boldsymbol{TT}^{-1}=\boldsymbol{I}$，有

$$H_{\mathrm{T}}(z)=[\boldsymbol{C}(z\boldsymbol{I}-\boldsymbol{A})^{-1}\boldsymbol{B}+\boldsymbol{D}]=H(z)$$

◀

8.4 系数量化误差

实际系统只能利用有限位来表示所设计的滤波器系数，量化误差是由于对滤波器系数

进行截尾或舍入处理产生的误差。有两种方法可降低量化误差带来的影响，第一种方法是增加位的数量。通常这种方法并不总是可行，因为位的增加总是有限度的。该方法将会增加成本和计算时间。第二种方法是采用受量化误差影响较小的实现结构，正如我们所见，对给定的系统选择合适的实现结构是滤波器设计过程中的关键部分。

为了更好地理解量化，这里考虑一个二阶系统的极点，其传输函数的分母为

$$D(z)=z^2+a_1z+a_2$$

一对复数共轭极点为

$$p_{1,2}=r\angle\pm\theta,\text{其中 } r=\sqrt{a_2},\theta=\arccos[a_1/(-2\sqrt{a_2})]$$

当量化 a_1 和 a_2 时，r 和 θ 值被限制取 z 平面上一系列离散的点上。图 8.24 给出了量化级为 6 位时的离散点位置。

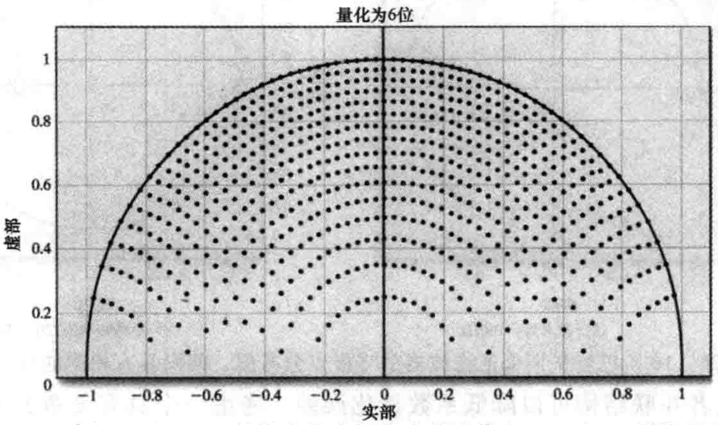


图 8.24 $z^2+a_1z+a_2=0$ 的根在极坐标系中的取值点，a_1 和 a_2 量化为 6 位

图 8.24 中对于特低频和特高频(对应的角频率 θ 很小或者很大)，可能的极点位置之间存在较大的空间，这意味着小相角或大相角的极点(或零点)对量化误差将非常敏感。

进一步考虑一个系统，其分母有三个实极点

$$D(z)=(z-p_1)(z-p_2)(z-p_3)=z^3+a_1z^2+a_2z+a_3$$

为确定每个极点 p_i 对系数 a_i 量化的敏感性，求其偏导数

$$-\frac{\partial p_1}{\partial a_1}(z-p_2)(z-p_3)=z^2$$

或者

$$\left.\frac{\partial p_1}{\partial a_1}\right|_{z=p_1}=\frac{-p_1^2}{(p_1-p_2)(p_1-p_3)}$$

同样，

$$\left.\frac{\partial p_2}{\partial a_1}\right|_{z=p_2}=\frac{-p_2^2}{(p_2-p_1)(p_2-p_3)}$$

$$\left.\frac{\partial p_3}{\partial a_1}\right|_{z=p_3}=\frac{-p_3^2}{(p_3-p_1)(p_3-p_2)}$$

极点 p_1 总的偏移为

$$\Delta p_1=\frac{\partial p_1}{\partial a_1}\Delta a_1+\frac{\partial p_1}{\partial a_2}\Delta a_2+\frac{\partial p_1}{\partial a_3}\Delta a_3$$

为不失一般性，对于 N 个实极点的情况有

$$\Delta p_k=\frac{-p_k^{N-j}}{\prod\limits_{\substack{i=1\\i\neq k}}^{N}(p_k-p_i)}\Delta a_j \tag{8.28}$$

从这个公式我们可以看出，为了使系数量化带来的极点偏移最小化，需要使得分母最大。分母为 p_k 和 p_i 间距离向量幅值的乘积，这就意味着需要让极点相互之间离得更远。经大量的数学推导，对于复数极点可以得到相同的结论[2]。

图 8.25 给出了两个 10 阶巴特沃斯低通滤波器，其中一个滤波器的抽样频率为另一个的 2 倍，其余的设计指标相同。结果显示抽样频率的加倍使得截止频率移动到更低的角度，导致极点更为集中，这些极点对系数的量化误差更为敏感。与多数认为的相反，抽样频率的上升会带来更多的噪声。

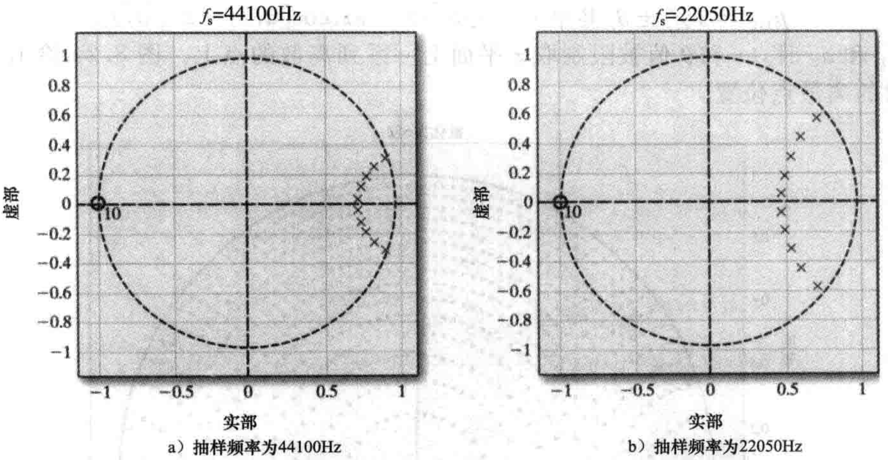


图 8.25 10 阶巴特沃斯低通滤波器的零极点分布图，两图具有相同的设计指标

采用级联或者并联结构可以降低系数量化误差。考虑一个具有复数共轭极点的二阶子系统，其传输函数的分母多项式如下

$$D(z) = z^2 + a_1 z + a_2$$

如果极点位于 $r\angle\pm\theta$，则有

$$a_1 = -(r/2)\cos(\theta), \quad a_2 = r^2$$

若量化误差很小，由链式法则(chain rule)可得

$$\Delta r = \frac{\partial r}{\partial a_1}\Delta a_1 + \frac{\partial r}{\partial a_2}\Delta a_2 = \frac{\Delta a_2}{2\sqrt{a_2}} = \frac{\Delta a_2}{2r}$$

由于 $\cos(\theta) = a_1/(2\sqrt{a_2})$，等式两边同时求导有

$$\Delta\theta = \frac{\partial\theta}{\partial a_1}\Delta a_1 + \frac{\partial\theta}{\partial a_2}\Delta a_2 = \frac{-\Delta a_1}{2r\sin(\theta)} + \frac{\Delta a_2}{2r^2\tan(\theta)}$$

当极点(或零点)接近原点时使得 r 非常小，或者说极点频率低使得 θ 非常小，此时二阶子系统具有较大的系数量化误差。该结果与式(8.28)相符，当 θ 增大时使得复数共轭极点间的距离增大。

下面的例题说明级联型二阶子系统如何降低量化误差的影响。

例 8-9 设计满足指标的巴特沃斯带阻滤波器

设计一个满足给定指标的巴特沃斯带阻滤波器，滤波器系数采用 12 位量化，利用 MATLAB 中的 freqz 函数画出滤波器的频率响应曲线，并与采用二阶子系统表示的滤波器频率响应曲线进行比较，其系数量化也采用 12 位。

设计指标：

抽样频率：11025Hz

通带、阻带波动：0.02

通带范围：0～600Hz，2500Hz～$f_s/2$

阻带范围：1100～2000Hz

解：下面的MATLAB代码生成带阻滤波器：

```
fs = 11025;
Rp = .02;RpDB = -20*log10(1-Rp);
Rs = .02;RsDB = -20*log10(Rs);
flp = 600;fls = 1100;fus = 2000;fup = 2500;
Wp = [flp fup]/(fs/2);
Ws = [fls fus]/(fs/2);
[N Wc] = buttord(Wp,Ws,RpDB,RsDB);
[num den] = butter(N, Wc, 'stop');
```

为了量化滤波器系数num和den为12位，首先采用quantize函数，将最大系数归一化至1，然后乘以2^{12}，通过舍入操作截断结果，除以2^{12}，最后再归一化。例如，当最大系数值为2时，量化某系数1.234567为12位字长，量化过程如下：

归一化：1.234567/2=0.6172835

乘以2^{12}=4096，得到：2528.393216

舍入至整数：2528.0

除以2^{12}(即4096)：2528/4096=0.6171875

归一化：2×0.6171875=1.234375

分子、分母系数量化完成后，可利用freqz函数得到频率响应并画出结果图。

为了得到二阶子系统的结果，利用MATLAB函数tf2sos将num和den分解为2次方程，求出零极点对。

```
sos = tf2sos(num, den);
```

经过该操作，sos为二阶子系统的矩阵，可得到每一个二阶子系统的幅度响应，通过相乘得到最终的结果。MATLAB程序如下：

```
num2 = sos(1:1,1:3); den2 = sos(1:1,4:6);
[H2 fd] = freqz(num2,den2,1024,fs);
H = abs(H2);
for i = 2:N
  num2 = sos(i:i,1:3); den2 = sos(i:i,4:6);
  [H2 fd] = freqz(num2,den2,1024,fs);
  H = H.*abs(H2);
end
plot(fd,abs(H2),'k');
```

图8.26给出了两种方法的结果。

由图8.26可见，当截尾量化二阶子系统的系数为12位时，并没有明显地改变滤波器的频率响应。然而，当将18阶滤波器看作两个系数都为12位的18阶多项式进行分析时，累计误差使得滤波器几乎无法使用。

通过在MATLAB中不断增加系数的量化级求解频率响应，多数情况下可以找到系统容许的量化级。图8.27给出了5阶低通椭圆滤波器的通带随着量化级由18位到10位的变化。

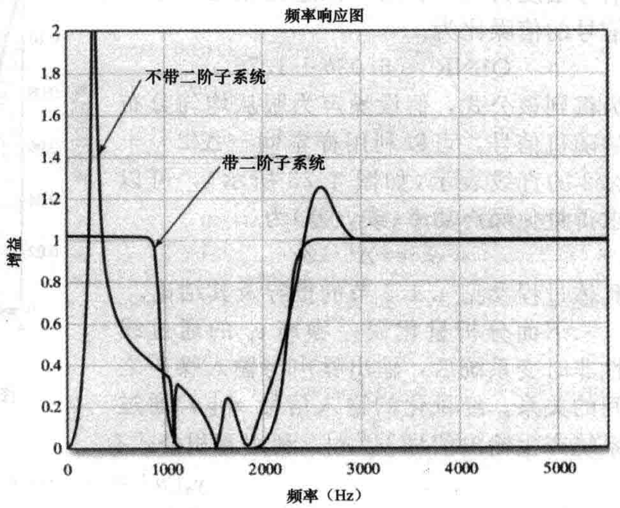


图8.26 18阶巴特沃斯带阻滤波器频率响应曲线，所有系数截尾为12位，其中一根曲线为二阶子系统表示的滤波器的频率响应

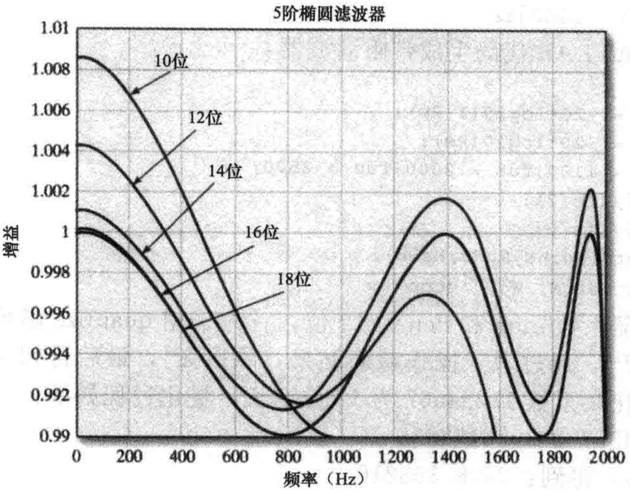


图 8.27　5 阶低通椭圆滤波器的通带随量化级的变化，其中最小的两个量化级得到的滤波器不满足设计指标

8.5 输入量化产生的输出误差

连续时间信号 $x(t)$经抽样后生成 $x(nT)=x[n]$，将其作为数字系统的输入并通过 A/D 量化生成量化信号 $x_q[n]$。信号 $x_q[n]$仅为信号 $x[n]$的近似，因而抽样信号与量化信号存在误差

$$\varepsilon_g[n] = x[n] - x_q[n]$$

其中，ε_g 称为粒状误差。

一个范围为$-A\sim+A$ 的信号，量化为 b 位，量化级之间的最小步长为

$$\Delta = 2A/2^b$$

粒状误差如图 4.22 所示，其总是小于或等于 Δ。图 8.28 画出了归一化的粒状误差随表示抽样值的位数变化的函数曲线。

第 4 章中，我们证明了如果输入正弦信号幅度为 A，利用 b 位进行量化，量化信号的信噪比为

$$\text{QSNR} \approx 6.02b + 1.76$$

为得到该公式，假设噪声为服从均匀分布的随机信号，可以利用样本间$-\Delta/2\sim+\Delta/2$ 的直线表示(如图 4.23 所示)，可以求得量化噪声功率(或方差)为

$$\sigma_n^2 = \Delta^2/12$$

具体过程参见 4.4.5 节的推导及其结论。

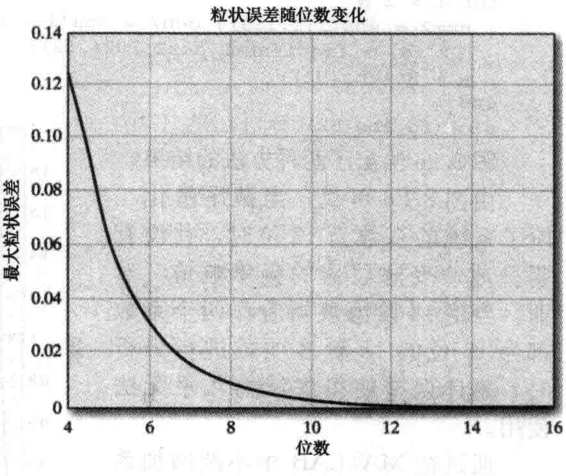


图 8.28　最大粒状误差随位数变化关系

下面分析量化误差噪声 ε_g 的通过线性非时变系统后，输出噪声与输入噪声之间的关系。经量化的输入信号 $x_q[n]$通过系统产生输出信号 $y_q[n]$，利用卷积公式有

$$y_q[n] = x_q[n] * h[n]$$

$x_q[n]$可以写为包含粒状误差 ε_g 的形式

$$y_q[n] = (x[n] - \varepsilon_g[n]) * h[n]$$

或

$$y_q[n] = x[n] * h[n] - \varepsilon_g * h[n]$$

令 $\varepsilon_o[n]=\varepsilon_g[n] * h[n]$，$\varepsilon_o[n]$为仅与粒状误差相关的系统输出。

$$y_q[n] = y[n] - \varepsilon_o[n]$$

由卷积公式，$\varepsilon_o[n]$可以求出

$$\varepsilon_o[n] = \sum_{k=-\infty}^{\infty} h[k]\varepsilon_g[n-k]$$

如果 $\varepsilon_g[n]$的方差(即功率) σ_g^2 已知，那么输出噪声的方差(功率)为

$$\sigma_o^2 = \sigma_g^2 \sum_{k=-\infty}^{\infty} h^2[k]$$

由以上公式，可以计算输出噪声的功率。

例 8-10　计算噪声功率

输入信号噪声方差为 $\sigma_n^2 = \Delta^2/12$，A/D 采用 8 位，信号范围为$-1\sim+1$，求解下列传输函数所描述系统的输出信号的噪声功率。

$$H(z) = \frac{0.0875(z+1)^2}{z^2 - 1.6z + 0.95}$$

解：由传输函数可以写出对应的差分方程

$$y[n] = 0.0875x[n] + 0.175x[n-1] + 0.0875x[n-2] + 1.6y[n-1] - 0.95y[n-2]$$

利用 MATLAB 计算脉冲响应，然后求其平方之和

```
fs = 1;T = 1/fs;
num = 0.0875*[1 2 1];
den = [1 -1.6 0.95];
[h nT] = impz(num, den, 200, fs);
figure(1);clf;
plot(nT, h);
h2 = sum(h.^2);
disp(h2);
```

以上程序求出脉冲响应 $h[n]$前 200 项的平方之和为

$$\sum_{k=-\infty}^{\infty} h^2[k] \approx \sum_{k=0}^{199} h^2[k] = 3.1936$$

由该公式有

$$\sigma_o^2 = \sigma_g^2 \sum_{k=-\infty}^{\infty} h^2[k] = 3.1936\Delta^2/12 = 0.26613\Delta^2$$

本例中 $\Delta=2/256=0.0078125$，$\Delta^2=0.00006104$。由量化带来的输出噪声功率为 1.624×10^{-5}。

利用 MATLAB 验证该结果，可通过对量化为 8 位的信号采用 filter 函数计算输出。选择输入频率为 0.075Hz 的正弦信号，输出幅度近似为 1。

以下命令可生成输入信号：

```
fsig = 0.075;
b = 8;
t = 0:T:1/fsig;
x = sin(2*pi*fsig*t);
```

利用 quant 函数量化输入信号为 8 位，函数说明参见附录 D。量化信号采用 filter 函数滤波。

```
xq = quant(x, b, -1, 1);
hold on;
plot(t, xq, 'r');
yq = filter(num, den, xq);
```

为确定量化噪声，需要知道未量化的 y 值，可将未量化输入信号通过滤波器来获得

```
y = filter(num, den , x);
```

输出信号的量化噪声功率为

```
N = length(t);
Pn = (sum((y-yq).^2))/N;
```

输出信号功率为

```
Ps = (sum(y.^2))/N;
```

从以上两个计算结果，可以计算输出信号的信噪比。

```
str = ['Pn = 'num2str(Pn) 'Ps =' num2str(Ps) '\n'];
fprintf(1, str);
str = ['QSNR =' num2str(10*log10(Ps/Pn))];
fprintf(1, str);
```

得到如下输出：

```
Pn = 1.5081e-005 Ps = 0.71174
QSNR = 46.7389
```

可见输出噪声功率 P_n 与计算值 σ_o^2 相当。输出量化信号的信噪比为46.7，与经验法则的每位约6dB相接近。

上面的方法可用来计算由量化带来的噪声功率，但该方法最多为合理近似。主要困难在于以上所做的假设得到的结果未必适用于实际信号，比如我们假设量化噪声服从均匀分布且与信号独立。缓慢变化的信号和低电平信号就不适用于以上假设。同样，若量化步长相对于信号较大，该假设也不成立。复杂信号如语音和音乐信号只有某些片段适用于以上假设。

8.6　乘积量化

差分方程的求解往往通过一系列的乘积和累加(MAC)运算而实现。由于两个 B 位的数字相乘生成 $2B$ 位的结果，在运算的某个阶段，累积结果必须再次缩减至 B 位。这可由截尾或者舍入操作实现。而无论哪种方法，都可能产生误差，此称为乘积量化误差。对于IIR数字滤波器，这种误差可被反馈回来造成潜在的误差累积。

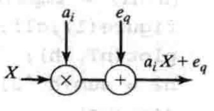


图8.29　乘积量化误差可以表示为每个乘积与误差之和

由于乘积量化误差是乘积的量化导致的结果，因而可把它表示为乘积与量化误差之和的形式，如图8.29所示。

图8.30显示了图8.23直接2型结构增加乘积量化误差模型后的结构图。

图8.30a对应的状态变量方程为

$$zv_1 = v_2$$
$$zv_2 = -a_2v_1 - a_1v_2 + x + e_0 + e_1$$
$$y = (b_2 - a_2b_0)v_1 + (b_1 - a_1b_0)v_2 + b_0x + b_0(e_0 + e_1) + e_3 + e_4$$

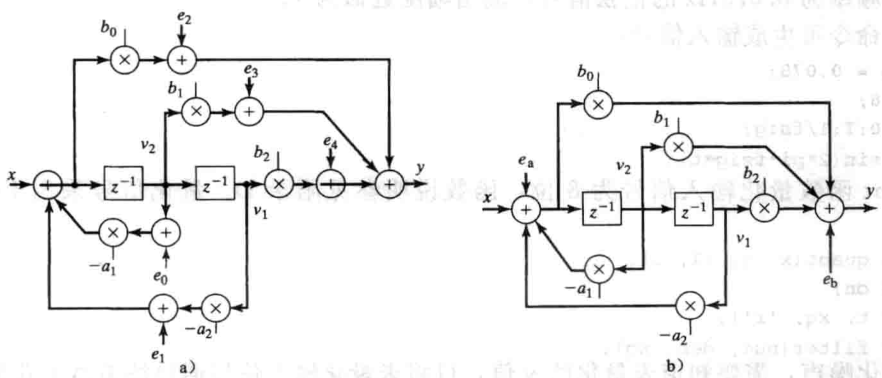


图8.30　带有乘积量化误差的直接2型结构，a与b功能上等同

如果令 $e_a = e_0 + e_1$，$e_b = b_0(e_0+e_1)+e_3+e_4$，则图 8.30a 可以转化为图 b 的线性图。需要注意的是噪声 e_a 来自分母且通过滤波器，噪声 e_b 来自分子且不通过滤波器。若用 p_n 表示由单个量化源产生的平均噪声功率，那么直接 2 型结构的噪声功率为

$$P_{df2} = Np_n\,|H(e^{j\Omega})|^2 + (M+1)p_n \tag{8.29}$$

其中，对于该系统 $N=M=2$。

求解 p_n 值需要对舍入误差特性进行假设。假设舍入误差在 $-q/2 \sim +q/2$ 范围内均匀变化，从而其概率密度函数可由图 8.31 描述。

$$p_{pdf}(x) = \begin{cases} 1/q, & -q/2 \leqslant x \leqslant +q/2 \\ 0, & \text{其他} \end{cases}$$

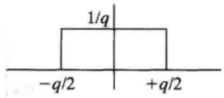


图 8.31 舍入误差的概率密度函数

误差的均方值，也称方差，定义为

$$p_n = \sigma^2 = \int_{-\infty}^{\infty} (x-\mu)^2 p_{pdf}(x)\,dx$$

式中，μ 是平均值或期望值，这里 $\mu=0$。

$$p_n = \int_{-q/2}^{q/2} x^2 (1/q)\,dx = \frac{q^2}{12}$$

将 p_n 代入式(8.29)中有

$$P_{df2} = (q^2/6)\,|H(e^{j\Omega})|^2 + q^2/4$$

一种最大限度地降低乘积舍入误差的方法是在一系列乘积和加法运算后只进行一次舍入操作，而非在每个乘积之后进行舍入操作。所以，这需要对乘积累加单元增加保护位或者以双精度进行运算，并在最后进行舍入。某些现代的 DSP 芯片和计算机系统正是如此，允许乘积累加单元在双倍宽度的寄存器中累加结果。

8.7 量化和抖动

抖动是一个违背直觉的概念，如有时可能需要在离散信号中加入噪声以降低来自 A/D 的量化误差。考虑图 8.32 中标有 S 的信号，该信号将被量化至标有 $(n-1)q$、nq 或者 $(n+1)q$ 的量化级中的一个。若未作特殊处理，则信号应量化为 nq。

若实现抖动，可在信号上增加一个随机数 r，这里 r 服从均匀分布，范围为 $-q/2<r<+q/2$。当信号 $S+r$ 的值大于或者等于 $nq+q/2$ 时，信号将落入下一个量化级，其值可以表示为 $(n+1)q$。如果 $S+r$ 低于 $nq+q/2$，可用量化电平 nq 表示。

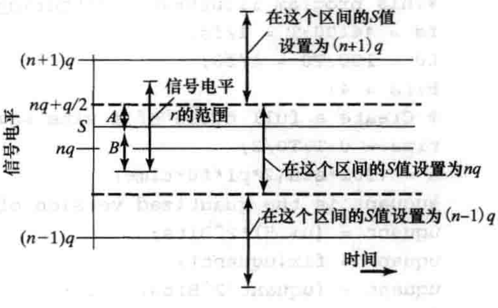


图 8.32 信号 S 量化前添加随机数 r 进行抖动

图 8.32 中，距离 A 为 $nq+q/2-S$，距离 B 为 $q/2$。$S+r<nq+q/2$ 的概率为 $(A+B)/q$。即

$$p(s+r<nq+q/2) = \frac{nq+q/2-S+q/2}{q} = \frac{(n+1)q-S}{q}$$

$S+r>nq+q/2$ 的概率为

$$p(S+r>nq+q/2) = 1-\frac{(n+1)q-S}{q} = \frac{S-nq}{q}$$

因为当 $S<nq+q/2$ 时，量化信号为 nq。当 $S>nq+q/2$ 时，量化信号为 $(n+1)q$，所以 S 的平均值为

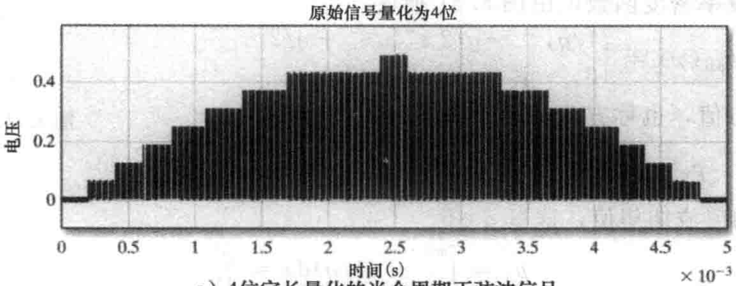
$$S\text{ 的均值} = \frac{(n+1)q-S}{q}nq + \frac{S-nq}{q}(n+1)q = S$$

下面举例说明抖动。

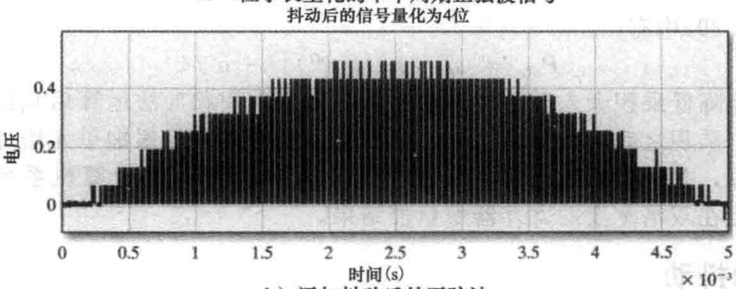
例 8-11 正弦信号的抖动

利用MATLAB生成半个周期的正弦波，频率为0.1kHz，抽样率为44.1kHz。利用4位进行正弦信号量化并画出曲线。在原始信号上增加抖动后再次利用4位进行量化，对比两者的区别。

解：下面的MATLAB程序产生0.1kHz的正弦波并利用4位进行量化。抖动信号加至正弦信号后再次利用4位进行量化。两次的曲线对比图如图8.33所示。



a) 4位字长量化的半个周期正弦波信号



b) 添加抖动后的正弦波

图 8.33

```
%Dither.m
%This program illustrates dithering using a sine wave
fs = 44100;T = 1/fs;
f0 = 100;T0 = 1/f0;
Bits = 4;
% Create a full cycle of a sine wave at f0.
time = 0:T:T0/2;
u = .501*sin(2*pi*f0*time);
%uquant is the quantized version of u.
uquant = (u+.5)*2^Bits;
uquant = fix(uquant);
uquant = (uquant/2^Bits) - .5;
%
figure(1);clf;
subplot(2,1,1)
stem(time, uquant, 'MarkerEdgeColor', 'w');
% rand produces a flat distribution between 0 and 1.
dither = (rand(1,length(u))-.5)/2^Bits;
ditherU = u + dither;
%dquant is the quantized dithered signal
dquant = (ditherU+.5)*2^Bits;
dquant = fix(dquant);
dquant = dquant/2^Bits - .5;
subplot(2,1,2)
stem(time,dquant, 'MarkerEdgeColor', 'w');
```

◀

抖动通过对信号增加半位的随机噪声而实现。信号越接近于下一个量化电平，编码为该量化电平的可能性就越高。尽管信号边缘出现大量“锯齿”，事实上这是对信号更好的一种平均表示。

8.8 溢出和缩放

对于DSP系统来说，溢出是指操作数不能被所用系统表示或者硬件寄存器没有足够的有效空间存储而造成的现象。溢出造成系统错误的输出，对于IIR数字滤波来说，通过反馈通道将这种错误再次输入系统，具有持续的影响。对任何系统输入足够大的数据都会造成溢出，而且(后面会看到)有些系统即使输入的是零也会产生溢出。

缩放是指通过增益因子调整信号以降低溢出的几率。经过适当比例缩放的系统，在输入定义范围内的信号时将不会出现溢出误差。由于浮点数系统的数值范围远远大于定点数系统的数值范围，因而溢出对于浮点运算的系统影响较小。以下绝大多数的讨论都基于定点系统的缩放进行。

对于构成级联型系统的二阶子系统，乘积缩放因子可以添加在二阶子系统之间以修改输入信号的范围，溢出将不会出现，如图8.34所示的系统。

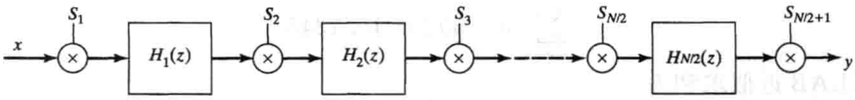


图8.34 具有缩放因子的级联的二阶子系统

需要指出的是，乘积缩放不会增加计算量，因为二阶系统传输函数要进行归一化处理，以使得 b_0 为1。而归一化因子和乘积缩放因子可以合并为一次乘法。

图8.35所示为单个二阶子系统的缩放因子计算。

图8.35的二阶系统中，标有 N_1 和 N_2 的两个节点有可能出现溢出误差，从输入到每个节点的传输函数为

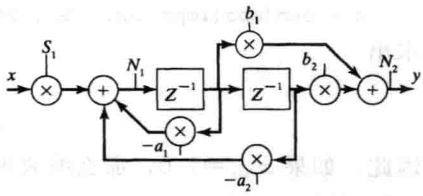


图8.35 单个二阶子系统中的缩放因子计算

$$H_1(z)=\frac{N_1}{X}=S_1\frac{z^2}{z^2+a_1z+a_2}$$

$$H_2(z)=\frac{N_2}{X}=S_1\frac{b_1z+b_2}{z^2+a_1z+a_2}$$

为了阻止溢出，需要

$$|N_i(k)|<1$$

其中，

$$|N_i(k)|=\left|\sum_{m=0}^{\infty}x(k-m)h_i(m)\right|\leqslant x_{\max}\sum_{m=0}^{\infty}|h_i(m)|\text{，这里，}i=1,2\text{。}$$

从而有，

$$S_i x_{\max}\leqslant\frac{1}{\sum\limits_{m=0}^{\infty}|h_i(m)|}\tag{8.30}$$

例8-12 确定缩放因子

为下面的二阶子系统确定合适的缩放因子

$$H(z)=\frac{0.2z^2+0.4z+0.2}{z^2-1.4z+0.8}$$

式中，$x_{\max}$设为1。

解：缩放之前，首先确定内部两个点的传输函数及其脉冲响应

$$H_1(z)=\frac{N_1}{X}=\frac{z^2}{z^2-1.4z+0.8}$$

$$h_1(m)=\delta(m)+1.4h_1(m-1)-0.8h_1(m-2)$$

$$H_2(z)=\frac{N_2}{X}=\frac{0.2z^2+0.4z+0.2}{z^2-1.4z+0.8}$$

$$h_2(m)=0.2\delta(m)+0.4\delta(1)+0.2\delta(2)$$
$$+1.4h(m-1)-0.8h(m-2)$$

利用MATLAB近似求和h_1。

```
num = 1;
den = [1 -1.4 0.8];
s = sum(abs(impz(num, den, 200))); %200 terms
```

求出

$$\sum_{m=0}^{200}|h_1(m)|=10.1345$$

利用MATLAB近似求和h_2。

```
num = [0.2 0.4 0.2];
den = [1 -1.4 0.8];
s = sum(abs(impz(num, den, 200))); %200 terms
```

求出

$$\sum_{m=0}^{200}|h_2(m)|=7.4286$$

因此，如果$x_{\max}=1.0$，那么缩放因子$S_1=1/10.1345=0.09867$。同理可求缩放因子$S_2=1/7.4286=0.13461$。◀

数学上，公式$\|x\|_p\equiv\left[\sum_{i=0}^{\infty}|x_i|^p\right]^{1/p}$称为向量$x$的$p$范数($p$-norm)，或者范数(norm)。

式(8.30)中，对于$\sum_{m=0}^{\infty}|h(m)|$，可以看出当$p$为1时有

$$\|h_i\|_1=\sum_{m=0}^{\infty}|h_i(m)| \tag{8.31}$$

式(8.31)称为L_1范数。利用L_1范数可以生成缩放因子以确保不出现溢出，使得输入保持在设定范围内。但是，L_1范数有一个相当保守的限制，缩放因子降低了信号的大小，但不能减小量化误差或者舍入误差的大小，所以过多地缩放会降低信号的信噪比。另外还有两种实现缩放的方法，其允许有可能的少量溢出，但不严格受L_1范数的限制。第一种方法为L_2范数方法，其表达式为

$$\|h_i\|_2=\sum_{m=0}^{\infty}(|h_i(m)|^2)^{1/2}$$

第二种方法为L_∞范数方法，通过求解当p趋向无穷时$\|x\|_p$的极限来实现。利用傅里叶变换，可以证明存在下列关系式

$$\|h_i\|_\infty=\max_{0<f<f_s/2}\{|H_i(f)|\}$$

L_∞范数可通过画出幅度响应曲线并寻找最大值求解。对于正弦输入可确保没有溢出。

在三种方法的限制方面，L_1 范数缩放受限最为苛刻，而 L_2 范数缩放受限最弱。

例 8-13 求解缩放因子

利用 L_2 范数和 L_∞ 范数两种方法求例 8-12 中二阶系统的缩放因子，并与例 8-12 的结果进行比较。

解： 例 8-12 中系统的传输函数和脉冲响应为

$$H_1(z)=\frac{N_1}{X}=\frac{z^2}{z^2-1.4z+0.8}$$

$$h_1(m)=\delta(m)+1.4h_1(m-1)-0.8h_1(m-2)$$

$$H_2(z)=\frac{N_2}{X}=\frac{0.2z^2+0.4z+0.2}{z^2-1.4z+0.8}$$

$$h_2(m)=0.2\delta(m)+0.4\delta(1)+0.2\delta(2)+1.4h(m-1)-0.8h(m-2)$$

利用 MATLAB 求解节点 1 处的 L_2 范数 L_{21} 为

```
num = 1;
den = [1 -1.4 0.8];
L21 = sum((abs(impz(num, den, 200))).^2);
```

节点 2 处的 L_2 范数 L_{22} 为

```
num = [.2 .4 .2];
den = [1 -1.4 0.8];
L22 = sum((abs(impz(num, den, 200))).^2);
```

整个系统的 L_2 范数为

```
L2 = max(L21, L22);
```

求出系统的 L_2 范数为 3.65，缩放因子为 1/3.65=0.3771。

计算两个节点的 L_∞ 范数，首先求出 H_1(z)和 H_2(z)对应的频率响应。设 f_s=2Hz 有

```
[H1 f] = freqz(num, 1, 1024, 2); %Node 1
[H2 f] = freqz(num, den, 1024, 2); %Node 2
```

系统的 L_∞ 范数 Linf 为

```
Linf = max(max(abs(H1)), max(abs(H2)));
```

求出的 L_∞ 范数为 5.7474，缩放因子为 1/5.7474=0.1740。

图 8.36 给出了系统三种缩放因子下的频率响应曲线。

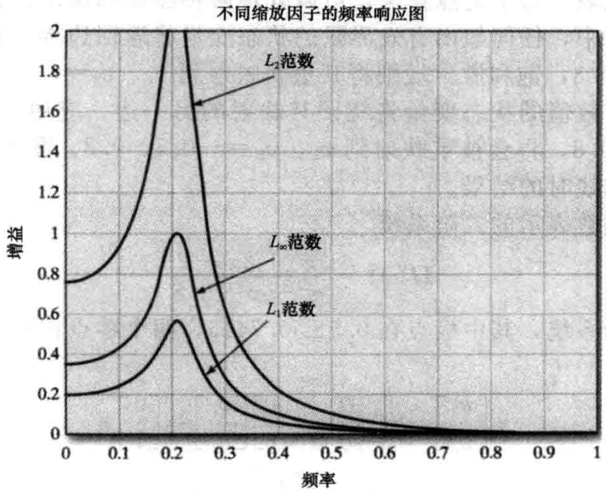


图 8.36 利用 L_1 范数、L_2 范数和 L_∞ 范数缩放得到的传输函数的频率响应曲线 ◀

8.9 极限环

对于一个稳定的数字滤波器，在输入为零且具有任意初始状态下，可能会产生一个短暂的输出，但最后一定会趋为零。在某些舍入误差或溢出情况下，系统会变为非线性，即使输入保持为零输出也会无限振荡。这种振荡通常称为极限环。

状态变量表示给出了一个产生并观察极限环的简单方法。对于二阶系统，状态变量表示为

$$zv = Av + Bx$$
$$y = Cv + Dx$$

式中，$v = \begin{bmatrix} v_1 \\ v_2 \end{bmatrix}$ 表示系统状态。如果输入为零，状态方程式变为(时域)

$$\begin{bmatrix} v_1(k+1) \\ v_2(k+1) \end{bmatrix} = A \begin{bmatrix} v_1(k) \\ v_2(k) \end{bmatrix} \tag{8.32}$$

式(8.32)称为轨迹方程(亦称状态转移方程)，表示在时间 $k\sim k+1$ 时的状态映射。因为仅有两个状态变量，所以可以在二维空间可视化状态转移的轨迹，如图 8.37 所示。简单起见，这里限制状态变量的有效值范围为－1～1。

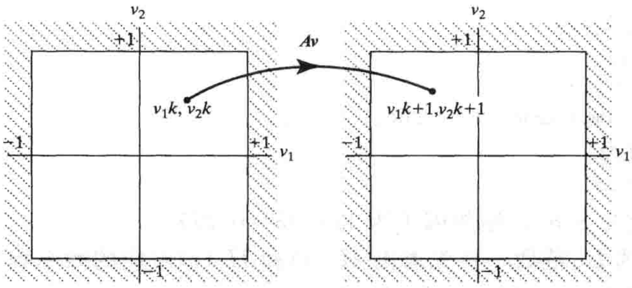


图 8.37 状态变量轨迹，输入为零，所以下一个状态由 Av 给出，其中阴影区域表示溢出

根据矩阵 A 中元素的值，轨迹方程可能会导致在下一状态出界即落入图 8.37 中阴影区域。因为不能表示这些阴影区域的数字，所以需要应用溢出规则来确定怎样将一个不合法状态映射回有效区域。对于定点算法，可以处理两种类型的溢出：饱和溢出和二进制补码溢出。在饱和溢出中，任何超出有效界限的状态变量将返回边界。例如，如果下一个状态为 v_1、v_2＝1.3、0.8，饱和溢出规则将其重新映射到 v_1、v_2＝1.0、0.8。在二进制补码溢出中，任何超出有效值的状态变量将绕回其映射的另一边。应用二进制补码溢出规则，状态 v_1、v_2＝1.3、0.8，将会被重映射到 v_1、v_2＝－0.7、0.8。图 8.38 给出了这两种溢出规则应用于正弦曲线时的结果。

考虑如下传输函数表示的二阶系统

$$H(z) = \frac{z^2 + 2z + 1}{z^2 - z + 0.8}$$

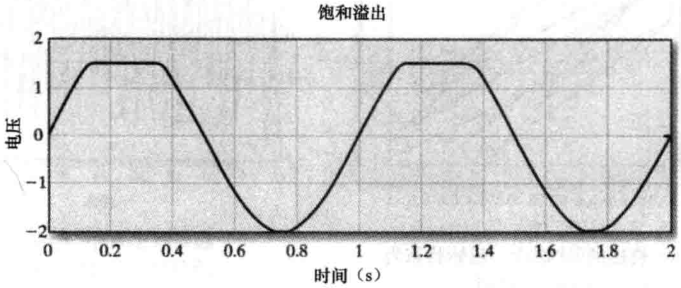
这是一个稳定的二阶系统，其中极点在 0.5±j0.7416，两个零点在 $z=-1$。状态变量表达式为：

$$z\begin{bmatrix} v_1 \\ v_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -0.8 & 1 \end{bmatrix}\begin{bmatrix} v_1 \\ v_2 \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} x$$

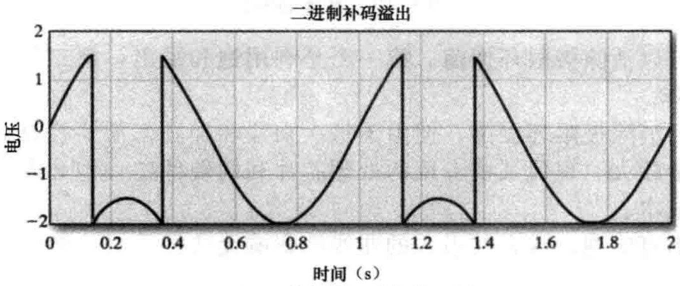
$$y = \begin{bmatrix} 2 & 3 \end{bmatrix}\begin{bmatrix} v_1 \\ v_2 \end{bmatrix} + [1]x$$

由式(8.32)得轨迹方程为，

$$\begin{bmatrix} v_1(k+1) \\ v_2(k+1) \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ -0.8 & 0 \end{bmatrix} \begin{bmatrix} v_1(k) \\ v_2(k) \end{bmatrix}$$



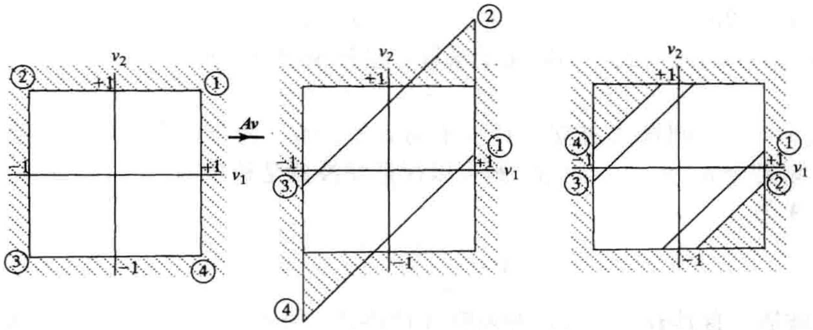
a）举例说明了超过1.5V的饱和溢出



b）二进制补码溢出的示例

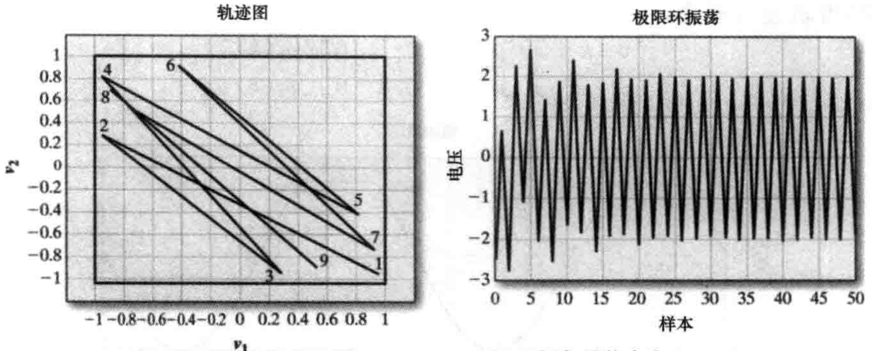
图 8.38　溢出示例

图 8.39 展示了所有可表示数值的区域是如何通过轨迹方程而映射。注意，两个三角区域代表溢出。当应用二进制补码溢出规则时，把溢出区域映射回可表示的数值区域。从这张映射图上可以清楚看到，任何接近拐角 2 的状态将通过溢出规则映射到接近拐角 4 的地方，反之亦然。因此，当这组方程被初始化为一个接近拐角 4 的状态，比如，在 v_1、v_2＝0.95、－0.95，其将交替地映射至接近拐角 2 和 4 两处，而不会衰减为零。结果是这两个状态变量将总是有非零数值，而这些值将导致一些输出不为零。输出变量 y 可通过 $y=\boldsymbol{Cv}$ 计算出来。图 8.40 给出了零输入，状态变量被初始化为 v_1、v_2＝0.95、－0.95 时的输出。



a）标准数空间表示　b）标准数空间映射经 $\boldsymbol{Av}$ 到平行四边形　c）应用二进制补码溢出规则，溢出区域被绕回可表示区域

图 8.39　通过轨迹方程映射的区域



a）轨迹的9次迭代，起始状态为 v_1、v_2=0.95、−0.95

b）起始状态为 v_1、v_2=0.95、−0.95，没有输入信号时的输出信号 y

图 8.40　系统的输出

有两种方法可以消除极限环振荡。第一个是利用饱和溢出，第二个是进行坐标变换并找到适当的 **A** 矩阵。

饱和溢出主要消除极限环振荡，但由于每次乘法和加法都需要检查是否溢出，所以其软件实现的运算量较大。现代某些专用的处理芯片利用硬件实现饱和溢出，从而占用较少或者不占用计算成本。

当饱和溢出不可实现，或者其引入的非线性影响无法接受时，修改 **A** 矩阵是一个可行的解决方法。

通过观察轨迹图中的溢出振荡，可以获知如何预测溢出振荡的发生。从轨迹图中可以看出，为了使系统输出衰减至 0，需使得当前状态比前一状态更靠近原点。所以，如果测量原点到当前状态的距离，可以利用该信息作为预测溢出振荡的依据。原点到状态 (v_1, v_2) 的距离为 $\sqrt{v_1^2+v_2^2}$。如果输入为 0，那么下一个状态为

$$\boldsymbol{Av}=\begin{bmatrix}a_{11}v_1+a_{12}v_2\\a_{21}v_1+a_{22}v_2\end{bmatrix}$$

所以，为了使得下一状态至原点的距离小于当前状态至原点的距离，需要满足

$$\sqrt{(a_{11}v_1+a_{12}v_2)^2+(a_{21}v_1+a_{22}v_2)^2}<\sqrt{v_1^2+v_2^2}$$

以保证不出现溢出振荡。

该式可变为

$$a_{11}^2v_1^2+2v_1v_2a_{11}a_{12}+a_{12}^2v_2^2+a_{21}^2v_1^2+2v_1v_2a_{21}a_{22}+a_{22}^2v_2^2<v_1^2+v_2^2 \tag{8.33}$$

当 $a_{12}=-a_{21}$，$a_{11}=a_{22}$ 时，式(8.33)易于求解，这种情况下可得

$$(a_{11}^2+a_{12}^2)(v_1^2+v_2^2)<v_1^2+v_2^2$$

如果 $(a_{11}^2+a_{12}^2)<1$，则该式成立。对于根为 $\alpha\pm j\beta$ 的二阶滤波器系统，稳定的条件为 $\alpha^2+\beta^2<1$，所以若令 $a_{11}=\alpha$，$a_{12}=\beta$，则可以保证滤波器是稳定的，并且不存在溢出振荡。滤波器的矩阵 **A** 为

$$\boldsymbol{A}=\begin{bmatrix}\alpha&\beta\\-\beta&\alpha\end{bmatrix} \tag{8.34}$$

由于极少的滤波器本身具有式(8.34)的矩阵 **A** 的形式，从而问题归结为如何将矩阵 **A** 进行变换以使得系统免于溢出振荡[3]。回顾式(8.24)～式(8.27)，可定义一个转换矩阵 **T** 并生成一个具有相同传输函数的系统，其矩阵 **A** 为

$$\boldsymbol{A}\leftarrow\boldsymbol{T}^{-1}\boldsymbol{AT}$$

为寻找矩阵 **T** 使得

$$T^{-1}AT = \begin{bmatrix} \alpha & \beta \\ -\beta & \alpha \end{bmatrix}$$

或者

$$\frac{1}{|T|}\begin{bmatrix} T_{22} & -T_{12} \\ -T_{21} & T_{11} \end{bmatrix}\begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}\begin{bmatrix} T_{11} & T_{12} \\ T_{21} & T_{22} \end{bmatrix} = \begin{bmatrix} \alpha & \beta \\ -\beta & \alpha \end{bmatrix} \tag{8.35}$$

矩阵 $\boldsymbol{A}$ 的特征值为特征方程 $|z\boldsymbol{I}-\boldsymbol{A}|=0$ 的根，因此有

$$\alpha = \frac{a_{11}+a_{22}}{2},\quad \beta = \frac{\sqrt{-(a_{11}-a_{22})^2-4a_{12}a_{21}}}{2}$$

代入式(8.35)，将有(经大量推导)

$$\boldsymbol{T} = \begin{bmatrix} -1/\beta & 0 \\ -\alpha/\beta & 1 \end{bmatrix},\quad \boldsymbol{T}^{-1} = \begin{bmatrix} -\beta & 0 \\ -\alpha & 1 \end{bmatrix} \tag{8.36}$$

利用式(8.24)～式(8.27)和式(8.36)，任何带有复根的二阶滤波器都可转换成无溢出振荡的滤波器。需要特别注意的是，该变换不能保证不发生溢出——除非原点到下一状态的向量长度小于原点到当前状态的向量长度。

例 8-14 系统变换避免溢出振荡

利用式(8.24)～式(8.27)和式(8.36)，将下面给出的系统变换成一个新的系统，使其免于溢出振荡，画出每个系统的零输入轨迹图，其中参数 v_1、$v_2=0.95$、-0.95。需要注意的是，该系统与图 8.38 中的系统相同。

$$z\boldsymbol{v} = \boldsymbol{A}\boldsymbol{v} + \boldsymbol{B}x$$
$$y = \boldsymbol{C}\boldsymbol{v} + \boldsymbol{D}x$$

式中，

$$\boldsymbol{A} = \begin{bmatrix} 0 & 1 \\ -0.8 & 1 \end{bmatrix},\quad \boldsymbol{B} = \begin{bmatrix} 0 \\ 1 \end{bmatrix},\quad \boldsymbol{C} = [2 \quad 3],\quad \boldsymbol{D} = [1]$$

解：首先求解 α 和 β，特征方程为

$$|z\boldsymbol{I}-\boldsymbol{A}| = 0 \text{ 或者 } z^2 - z + 0.8 = 0$$

该方程的共轭复根为 $z=0.5\pm \mathrm{j}0.7416$，所以 $\alpha=0.5$，$\beta=0.7416$。由于 $\alpha^2+\beta^2=0.8$，系统保持稳定。转换矩阵 $\boldsymbol{T}$ 为

$$\boldsymbol{T} = \begin{bmatrix} -1/\beta & 0 \\ -\alpha/\beta & 1 \end{bmatrix} = \begin{bmatrix} -1.348 & 0 \\ 0.6742 & 1 \end{bmatrix},\quad \boldsymbol{T}^{-1} = \begin{bmatrix} -0.7418 & 0 \\ 0.5 & 1 \end{bmatrix}$$

由式(8.24)～式(8.27)可得

$$\boldsymbol{A} \leftarrow \boldsymbol{T}^{-1}\boldsymbol{A}\boldsymbol{T} = \begin{bmatrix} -0.7418 & 0 \\ 0.5 & 1 \end{bmatrix}\begin{bmatrix} 0 & 1 \\ -0.8 & 1 \end{bmatrix}\begin{bmatrix} -1.348 & 0 \\ 0.6742 & 1 \end{bmatrix} = \begin{bmatrix} 0.5 & -0.7418 \\ 0.7418 & 0.5 \end{bmatrix}$$

$$\boldsymbol{B} \leftarrow \boldsymbol{T}^{-1}\boldsymbol{B} = \begin{bmatrix} -0.7418 & 0 \\ 0.5 & 1 \end{bmatrix}\begin{bmatrix} 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

$$\boldsymbol{C} \leftarrow \boldsymbol{C}\boldsymbol{T} = [2 \quad 3]\begin{bmatrix} -1.348 & 0 \\ 0.6742 & 1 \end{bmatrix} = [-4.7186 \quad 3]$$

$$\boldsymbol{D} \leftarrow \boldsymbol{D} = [1]$$

下面的 MATLAB 程序使用 $\boldsymbol{A}$ 矩阵画出了轨迹图。显示了轨迹的前 50 点并按顺序相连，如图 8.41 所示。

```
num = [1 2 1];den = [1 -1 .8];
r = roots(den);
alpha = real(r(1,1));beta = imag(r(1,1));
A=[0 1;-0.8 1];
B=[0;1];
C=[2 3];
D=[1];
T = [-1/beta 0;-alpha/beta 1];
At = T^-1*A*T;
Bt = T^-1*B;
Ct = C*T;
figure(1);clf;
axis([-1.2 1.2 -1.2 1.2]);
line([1 -1],[1 1]);line([-1 -1],[1 -1]);
line([-1 1],[-1 -1]);line([1 1],[-1 1]);
x = [0.95;-0.95];count = 0;
while(((v(1,1) ~= 0) & (v(2,1) ~= 0)) & count < 50)
 Vnew = At*v;
 % Use twos complement overflow
 Vnew = twosover(Vnew);
 x = [v(1,1) Vnew(1,1)];
 y = [v(2,1) Vnew(2,1)];
 line(x,y,'Color','blue');
 v = Vnew;
 count = count + 1;
end
```

```
%twosover
function s = twosover(x);
  for i = 1:length(x)
      if (x(i)>=1)
        x(i) = x(i) - 2;
      else
        if (x(i) < -1)
          x(i) = x(i) + 2;
        end
      end
  end
s = x;
```

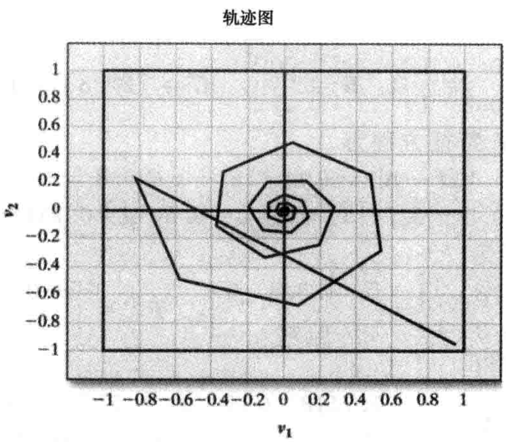


图 8.41 变换后系统的轨迹图，系统与图 8.40 所示系统相同，变换 $\boldsymbol{A}$ 矩阵消除了由二进制补码溢出造成的周期振荡

注意，图 8.41 中的初始状态为(0.95、−0.95)，轨迹呈螺旋形趋于零点，可与图 8.40中的振荡轨迹作对比。◀

Ebert[4]等人和 Roberts、Mullis[3]已经证明，若二阶系统的 $\boldsymbol{A}$ 矩阵为 $\boldsymbol{A}=\begin{bmatrix}a_{11} & a_{12}\\ a_{21} & a_{22}\end{bmatrix}$，如果

$$a_{12}a_{21}\geqslant 0$$

或者

$$a_{12}a_{21}<0 \text{ 并且 } |a_{11}-a_{22}|+|\boldsymbol{A}|<1$$

则系统不存在溢出振荡。另有其他方法解决溢出和溢出振荡问题，一个简单的方法就是利

用饱和溢出特征代替二进制补码。在饱和溢出中，所有溢出导致数值被设置成最大值。但在实现饱和溢出处理的过程中，将需要额外的硬件资源或者额外的软件开销去检测每一次计算和校正溢出。

同样，某些滤波器的实现中本身就不会出现溢出振荡，如格型结构。

8.10 微控制器 DSP

微控制器与通用的计算机有着较大的区别，它通常由单块芯片构成，包括只读程序存储器、小容量的读/写存储器、特殊功能的 I/O(输入/输出)器件，如键盘控制器、A/D 和 D/A。微控制器出现于 20 世纪 70 年代中期，广泛应用在专用计算机系统中。由于计算机行业逐渐向着更小、更快、更低功耗方向发展，微控制器也不例外。在过去的 20 年里，微控制器的计算速度已足够执行某些 DSP 功能，越来越多的应用随之出现，如汽车中的 DSP 处理器。

8.10.1 微控制器 DSP 的特点

差分方程的实现需要一系列乘法器和加法器，乘法器总以常量乘以一个变量。只读存储器可以用来存储程序和常量，读/写存储器用于存储变量。为了处理模拟信号，需要 A/D和 D/A 等接口。这些硬件的需求不是那么苛刻，因为某些微控制器拥有匹配的硬件特征。但是，存在两个严格的要求：精度和速度。抽样定理设置的抽样速率是至关重要的参数，因为在一个抽样周期内，需要完成从模拟-数字转换、到差分方程求解以及数字-模拟转换的全部过程。可以接受的噪声量级取决于精度要求。虽然存在某些例外，但总体而言，8 位微控制器将不适用于整个 DSP 应用系统，但某些极低量级的系统除外。下面列出了微控制器应用于 DSP 时所期望的特征。

1. 精度——精度指微控制器中的数据路径和 ALU 环的宽度。对于多数的 DSP 应用，16 位可满足要求。应用浮点型运算将更好，但不是必需的。

2. 速度——对于多数的微控制器来说，模拟-数字转换和乘法是最慢的运算。运算时间对于不同的问题各不相同，但是遵循一个粗略的经验定理

$$A/D\text{ 时间}+N\times(\text{乘法时间}+\text{加法时间})<0.8T$$

其中，N 为乘法运算的数量，T 为抽样周期。

3. MAC 单元——仅有少数高端微控制器中内置有乘法及累加(MAC)单元，这些硬件单元可以在一次运算中同时执行乘法、加法和更新指针的操作。

4. 存储器——多数微控制器中的存储器对 DSP 应用是足够的，此要求不高。例如，一个 FIR 数字滤波器需要 100 个延迟变量，每个变量占用 32 位，则共需 400 字节，按照当今的标准程序存储器也不需要占用太多的存储空间。

5. A/D 转换——许多微控制器上的 A/D 转换器与数据的宽度不同，由于片上 A/D 转换需要整合模拟和数字量，这将占用相当大的存储空间。因此，16 位或 32 位的微控制器上常出现 10～12 位的转换器。虽然位数越大越好，但是 10～12 位对多数应用来说已足够。

6. D/A 转换——某些微控制器已经内置了 D/A 转换器，或者外扩也很容易。但多数提供脉宽调制(Pulse-Width Modulated，PWM)输出，利用低通滤波器即可将其转换成模拟信号。鉴于精度问题，D/A 或 PWM 的输出需要匹配 A/D。

7. 定时——定时器对于保持抽样时间，并保证时间间隔的一致性至关重要。中断系统不是必要的，但绝大多数的微控制器都有中断系统，且应用于 A/D 和定时器。

8. 可预测的执行时间——几乎所有的微控制器都满足这个需求，除了某些高端的 CPU 之外。此外，带有多个管道和其他形式的平行结构的芯片不可预测执行时间。

多数微控制器没有专用 DSP 系统的某些特征，如溢出饱和运算、环形缓冲器以及零开销循环机制等。

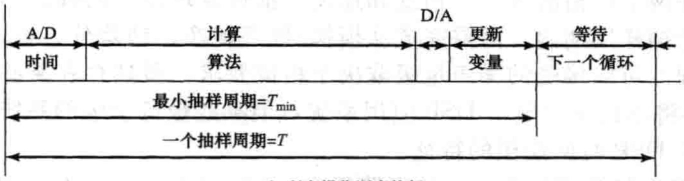
8.10.2 C语言实现

运行于微控制器上的DSP算法的伪代码步骤如下所示：

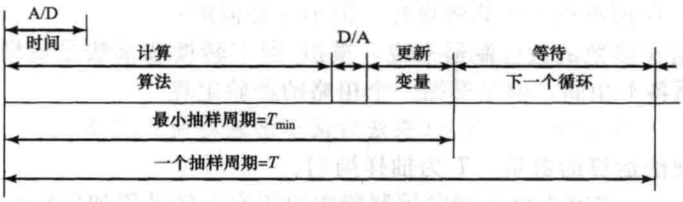
```
Initialize variables
Loop forever
    CALL AtoD(uIn)
    Calculate algorithm for one iteration
    CALL DtoA(yOut)
    Update variables for next iteration
    Wait until next sample period
End loop
```

该循环程序必须在每个抽样周期内执行一次循环。命令行“Calculate algorithm for one iteration”可以像一个单独的差分方程一样简单或者包含许多操作实现多级的级联滤波器。命令行“Wait until next sample period”保证每一次循环的执行时间与滤波器的抽样时间相同。如果循环中所有指令按顺序执行，则时序图将如图 8.42a 所示。

图 8.42b 中，模-数的转换与计算算法并行。这种情况下，计算算法需要用到的只是上一个周期的模-数转换值，当前的模-数转换值是为下一个周期准备。数-模转换时间也可以重叠，但是一般的数-模转换时间非常短，占用时间很少。



a）所有操作顺序执行



b）某些运算可并行实现

图 8.42 典型的DSP算法实现的时序图

上一节讨论了多种不同的实现结构，本节讨论这些结构的C语言实现。

8.10.3 FIR滤波器的C语言实现

直接型。图 8.6 中的直接型 FIR 滤波器实现非常适用于紧密循环，其主要依靠存储于数组中的延迟变量。图 8.43 给出了具有对称系数的 12 阶 FIR 低通滤波器的 C 语言实现。

图 8.43 中，第 1～4 行命令声明滤波器系数为常数。由于滤波器是对称的因而仅需一半的系数。第 8 行设置一维数组用于保存单位延迟的输出。第 10 行中的模-数转换函数和第 15 行的数-模转换函数没有给出。第 11 行假设我们已采用了 10 位的模-数转换，并将所有的输入转化为 0～1 范围的浮点数。第 13～14 行处理差分方程中的所有乘法和加法运算。第 16～17 行平移所有变量为下一个周期做准备。

此实现的主要优势在于该程序易于编写和修改，对称性提高了计算的效率。通过分解第 12～13 行的循环，分别编写乘法和加法将使得程序运行加快。

FIR 转置型。图 8.7 中的转置型实现可采用近似的结构实现，但其在利用系数的对称性方面较为困难。该问题留作练习(见习题 8.28)。

```
1   const float b[] = {-0.00725966, 0.04014696, 0.02354041,
2                      -0.08357119, -0.02859982, 0.31030209,
3                      0.53138928
4                     };
5   int main()
6     {int i, xInt;
7      float y;
8      float x[13];
9      while(1)
10       {//Do A/D to xInt
11        x[0] = (float) (xInt)/1024.0;
12        for(i=0;i<6;i++)
13            y += b[i]*(x[i] + x[12 - i]);
14        y += b[6]*x[6];
15        //Send y to D/A
16        for(i=12;i>0;i--)
17            x[i] = x[i-1];
18        //Wait until next cycle
19       }
    }
```

图　8.43

级联 FIR 子系统。图 8.9 中的级联型 FIR 实现表明，每个子系统实现一个二阶差分方程，形式为

$$y_s = b_{s0}x_{s0} + b_{s1}x_{s1} + b_{s2}x_{s2}$$

其中 s 为二阶子系统的编号，y 为输出，x 为输入及其延迟。由于是级联形式，每个二阶子系统的输出变为下一个二阶子系统的输入。一般而言，方程可变为

$$y_0 = b_{00}x_{00} + b_{01}x_{01} + b_{02}x_{02} \qquad x_{10} = b_{00}x_{00} + b_{01}x_{01} + b_{02}x_{02}$$

$$y_1 = b_{10}y_0 + b_{11}x_{11} + b_{12}x_{12} \qquad x_{20} = b_{10}x_{10} + b_{11}x_{11} + b_{12}x_{12}$$

…　　　　…

还可以写成

$$y_m = b_{m0}y_{m-1} + b_{m1}x_{m1} + b_{m2}x_{m2} \qquad x_{m+1,0} = b_{m0}x_{m0} + b_{m1}x_{m1} + b_{m2}x_{m2}$$

$$y_{out} = x_{m+1,0}$$

双下标建议在紧密循环中采用二维数组，图 8.44 给出了利用例 8-2 中的汉明窗实现 9 阶 FIR 滤波器的 C 语言代码。

图 8.44 中有两个二维数组，1～6 行定义了一个二维数组用于保存常数，第 10 行定义了一个二维数组用于保存延迟变量。两个二维数组中，第一个标号表示二阶子系统的编号为 0～4，第二个标号表示属于那个二阶子系统的项。第 12 行输入信号到 A/D，模-数函数开始转换并等待完成。13 行将整数转换为 0～1 范围的浮点数。第 14～17 行计算滤波器对应的各个二阶子系统，并将一个二阶子系统的输出作为下一个子系统的输入。第 18～19 行将输出送至 D/A，第 20～23 行平移每个二阶子系统的变量为下一个循环做好准备。

这段代码紧凑并易于修改，二阶子系统的重新排列不会影响结果。缺点在于循环的总体结构，以及二维数组做了许多不必要的系数为 1 的乘法运算。同样，常数存储在存储器中，每次乘法需要经过索引和重新取回。

```
1   const float b[5][3] = { {0, -0.003974, 0.021105},
2                           {1, 0.811702, -0.188298},
3                           {1, 1.677041, 1.986063},
4                           {1, 1.284306, 1.00000},
5                           {1, 0.844405, 0.503509}
6                         };
7   int main()
8     {int i, xInt;
9     float y;
10    float x[6][3];
11    while(1)
12      {//Do A/D to xInt
13       x[0][0] = ((float)xInt/1024.0;
14      for(i=0;i<5;i++)
15        {x[i+1][0] = b[i][0]*x[i][0] +
16                b[i][1]*x[i][1] + b[i][2]*x[i][2];
17        }
18      y = x[5][0];
19      //Send y to D/A
20      for(i=0;i<5;i++)
21        {x[i][2] = x[i][1];
22         x[i][1] = x[i][0];
23        }
24      //Wait until next cycle
25      }
26    }
```

图 8.44　9 阶 FIR 滤波器级联子系统的 C 代码

FIR 并联或多相实现。FIR 滤波器的并联或多相实现如图 8.10 所示。每个二阶子系统都可以看作独立的直接型滤波器。多相结构允许每个二阶子系统使用多个独立的处理器来实现，且所有的处理器并行运算，共享同步的时钟信号。

8.10.4 IIR 滤波器的 C 语言实现

正则直接 2 型。对于 IIR 数字滤波器，如果滤波器的阶数小于 10，则图 8.14 所示的正则直接 2 型将在实现中经常用到，差分方程为

$$y[n]=b_0x[n]+b_1x[n-1]+\cdots+b_Mx[n-M]-a_1y[n-1]-a_2y[n-2]-\cdots-a_Ny[n-N] \tag{8.37}$$

由图 8.14 也可以得到下列表示式

$$w[n]=x[n]-a_1w[n-1]-\cdots-a_Nx[n-N] \tag{8.38}$$

$$y[n]=b_0w[n]+b_1w[n-1]+b_2w[n-2]+\cdots+b_Mw[n-M] \tag{8.39}$$

这两个表示式没有减少乘法器的数量，但它们可以减少需要存储的变量的数量。

在图 8.45 所示的程序中，系数存储于数组中，使得在一个紧密循环中求解差分方程成为可能(第 17～20 行)。对于多数微控制器，如果消除循环直接计算差分方程可以使该程序运行速度更快。

级联的二阶 IIR 子系统。可以利用式(8.38)和式(8.39)实现级联的二阶子系统。这种情况下，每个二阶子系统的输出为下一个二阶子系统的输入。图 8.46 给出了级联的二阶

子系统的 C 语言代码。每个系数带有两个数字下标，其中第一个下标表示二阶子系统序号，第二个下标表示变量位置。初始输入变量为 x，输出为 y40。

通常情况下，大部分系数的值为 1，少数的值为 0，所以可以消除这些系数以减少不需要的乘法运算。

```
1   const float b[] = {0.0421840413, 0.0367065554, 0.0974706124,
2                       0.0749573553, 0.0974706124, 0.0367065554,
3                       0.0421840413};
4   const float a[] = {1.0000000000, -2.3202692892, 3.5758572131,
5                      -3.2514128312, 2.0435521083, -0.7592600647,
6                       0.1435326349};
7   int main(void)
8       {unsigned int xInt;
9        float x, y, w[7];
10       int i;
11       while(1)
12        {//A/D to xInt
13         w[0] = (float)xInt/1024;
14         y = 0;
15         for(i=1;i<7;i++)
16           w[0] -= a[i]*w[i];
17         for(i=0;i<7;i++)
18           y += b[i]*w[i];
19         //Send y to D/A
20         for(i=6;i>0;i--)
21           w[i] = w[i-1];
22         //Wait until next cycle
23        }
24      }
```

图 8.45　正则直接 2 型 6 阶 IIR 滤波器的 C 代码

```
//first section
w10 = x - a11*w11 - a12*w12;
y10 = b10*w10 + b11*w11 + b12*w12;
// second section with 'y10' as the input, and 'y20' as the output
w20 = y10 - a21*w21 - a22*w22;
y20 = b20*w20 + b21*w21 + b22*w22;
//third section with 'y20' as the input, and 'y30' as the output
w30 = y20 - a31*w31 - a32*w32;
y30 = b30*w30 + b31*w31 + b32*w32;
//fourth section with 'y30' as the input, and 'y40' as the output
w40 = y30 - a41*w41 - a42*w42;
y40 = b40*w40 + b41*w41 + b42*w42;
```

图 8.46　级联二阶子系统的实现代码，此为 8 阶椭圆滤波器，由 4 个二阶子系统组成，每个二阶子系统使用直接 2 型实现

IIR 格型滤波器子系统。带有极点和零点的 IIR 格型滤波器可由式(8.8)、式(8.9)和式(8.10)实现。图 8.21 中，m_i 从右侧开始编号，但求解从左侧开始。下面的 C 代码实现了 6 阶 IIR 椭圆格型数字滤波器，其中格型系数和差分方程项存储于数组之中。

在图 8.47 中，每个子系统的格型结构中有两次乘法运算。某些格型结构需要更少的乘法[5]，但一般来说，其计算效率低于直接型结构。

```
//Elliptic order = 6
//Numerator 0.04661  0.05475  0.12032
//             0.10798  0.12032  0.05475  0.04661
//Denominator  1.0000  -2.0750  3.1846
//             -2.8023  1.7827  -0.6637  0.13085
const int N = 6;
//Lattice m coefficients
const float m[] = {-0.38280,  0.96366,  -0.60047,
                   0.69268,  -0.39918,  0.13085};
//Lattice c coefficients
const float c[] = {-0.03246,  -0.00411,  0.10488,
                  0.26971,  0.27826, 0.15146,  0.04661};
int main (void)
  {unsigned int xInt;
   float x, yOut;
   float w[] = {0, 0, 0, 0, 0, 0, 0, 0};
   int s;
   while(1)
    {while ((xInt & 0x80000000) == 0)
     //A/D to xInt;
     y[0] = (float)xInt/1024;
     for(s=1;s<7;s++)
       {y[s] = y[s-1] - m[N-s]*w[s+1];
        w[s] = m[N-s]*y[s] + w[s+1];
       }
     w[7] = y[6];
     yOut = 0;
     for(s=1;s<7;s++)
       yOut += c[s]*w[s];
     //Send y to D/A
     //Wait for next cycle
    }
  }
```

图 8.47　二阶格型结构的软件实现

格型结构最主要的优点在于，其高阶滤波器对量化误差的敏感程度较小。

状态空间二阶 IIR 子系统。状态空间表示几乎可应用于所有的滤波器实现。其在溢出、缩放和量化误差方面有着许多优点，但一般不会提高计算效率。图 8.48 给出了一个 6 阶椭圆滤波器的 C 代码实现，其中使用状态空间表示每个级联的二阶子系统。

8.10.5 速度优化

在利用微控制器实现 DSP 算法时，速度优化常常至关重要。因为微控制器速度一般低于专用 DSP 处理器，且微控制器的硬件不是专门为 DSP 而设计的。下面是在进行速度

优化时需要考虑的一些要素。

1. 利用汇编语言。虽然编译器和编程器各不相同，但汇编语言代码往往可以使得某些算法运行速度更快——几乎两倍于高级语言实现的相同算法。汇编语言的缺点在于代码的可读性差，并且很难正确编写。一个好的策略是使用 C 语言实现算法，对关键部分使用汇编语言编程。该方法可以合理地平衡编程工作量与算法速度。

```
//SOSExample.c
//second order sections
//0.041953  0.054929  0.041953  1.000000  -0.935199  0.296056
//1.000000  -0.091121  1.000000  1.000000  -0.986939  0.728775
//Section 1
const float A1[4] = {0, 1.0,  -0.2961,  0.9352};
const float B1[2] = {0,1};
const float C1[2] = {0.0295,  0.9352};
const float D1 = 0.0420;
//Section 2
const float A2[4] = {0, 1.0,  -0.7288,  0.9869};
const float B2[2] = {0,1};
const float C2[2] = {0.2712,  0.8958};
const float D2 = 1;
int main(void)
  {unsigned int xInt;
   unsigned int dtoaOut;
   float x;
   float Zx11, Zx12, Zx21, Zx22, x11 = 0, x12 = 0, x21 = 0, x22 = 0;
   float y1, y2;
   while(1)
     {//A/D to xInt
      x = (float)xInt/1024.0;
      //Section 1
      Zx11 = A1[0]*x11 + A1[1]*x12 + B1[0]*x;
      Zx12 = A1[2]*x11 + A1[3]*x12 + B1[1]*x;
      y1 = c1[0]*x11 + c1[1]*x12 + D1*x;
      //Section 2
      Zx21 = A2[0]*x21 + A2[1]*x22 + B2[0]*y1;
      Zx22 = A2[2]*x21 + A2[3]*x22 + B2[1]*y1;
      y2 = C2[0]*x21 + C2[1]*x22 + D2*u;
      //Output
      //Send y to D/A
      x11 = Zx11;
      x12 = Zx12;
      x21 = Zx21;
      x22 = Zx22;
      //Wait for next cycle
     }
  }
```

图 8.48　6 阶椭圆滤波器的状态空间实现

2. 尽可能并行实现。在大多数情况下，当前周期进行计算的过程中同时实现A/D转换是可行的。级联和并联滤波的子系统同样提供了采用不同处理器并行计算的机制。

3. 采用整型运算。浮点型运算使得程序员免于考虑溢出和缩放，但相比整型实现相同操作，浮点型乘法占用更长的运算时间。对于微控制器而言，若利用软件处理浮点运算，而不是利用专用浮点单元进行运算，计算时间将更加难以承受。

4. 理解处理器架构。对比条件不满足，条件满足下的执行分支往往占用更长的计算时间。这是由于处理器必须提取另一个地址而非执行下一条指令。如果可能，重新组织逻辑以使得多数时间下的条件转移不发生。同样，多数微控制器分割存储空间以实现就近寻址，以减少执行时间。从外部存储器中提取变量可能需要占用两倍至三倍从片上存储器中提取变量的时间。

5. 优化乘法运算。乘法是差分方程求解中的核心运算，且往往是多数微控制器执行指令中最慢的运算。

- 避免与1、0的乘法运算。
- 不要做两个常数的乘法，可以将两个常数的乘积作为第三个常数，提前放置于程序之中。
- 如果乘积结果在程序中使用多次，可以只做一次乘积并将结果存于变量之中便于再次使用。
- 若有可能，利用算术平移方法实现2的幂次乘法运算，例如，与10相乘等同于乘以(8+2)，这样便可以利用平移和加法运算实现。

6. 尽可能利用对称性。数字先求和再乘积的运算顺序要优于数字先相乘再求和的运算顺序。例如，$(a+b)*c$ 比 $a*c+b*c$ 运算快。

7. 展开循环。执行10遍相同的操作速度要快于将其运行于一个10次循环之中。原因在于循环的机制，这也说明了速度和空间之间的平衡。这种情况下，通过去除循环，以更多的程序存储空间换取更快的运算速度。在某些高端的专用DSP芯片中，通过某些硬件使得循环机制变得微不足道。而在某些通用的处理器中，循环被分解到缓存以便更多的指令可同时执行。

8. 数据平移是耗时的操作。尝试使用指针平移替代数据平移。

习题

概念题

1. 利用MATLAB中的ss2sos函数，将状态空间形式转换成二阶子系统并用矩阵排列，每个二阶子系统的分子和分母一起排列为矩阵的一行。按顺序排列的行中最后一行是极点最靠近单位圆的二阶子系统。
   (a) 二阶子系统的排列顺序为什么重要？
   (b) 第一个二阶子系统(接收原始输入信号)能否具有距单位圆最近或最远的极点？为什么？
2. 简短地讨论饱和溢出和二进制补码溢出技术的优缺点。
3. 在数字滤波器应用中，无溢出并不代表任何情况下都不会出现溢出。请简短解释一个无溢出的滤波器在什么情况下真正地不会出现溢出。
4. 输入量化和系数量化噪声如何影响滤波器的相移？说明你的答案。
5. 请解释系数量化产生的误差与系统实现中的抽样率有怎样的关系。
6. 举出至少一种方法由经验测量确定在标准台式机上的常数与变量乘积以及两个变量相加的时间。
7. 利用数学方式证明IIR滤波器比相应的FIR滤波器对量化噪声更加敏感。
8. 只需通过增加系数和变量的位数即可降低量化误差，为什么在许多实际系统中并不采用此方法？请给出两个原因。
9. 列出数字滤波器算法应用的基本步骤，并说明(近似地)每个步骤的运算数量如何随着滤波器阶数而变化。

10. 如果某数字滤波器的算法在多任务计算机系统中执行，如何确定抽样时间？
11. 假设一个数字滤波器实时运行，已知其输入和输出，但滤波器或算法未知。
    (a) 描述至少两种方法用于确定滤波器的抽样率。
    (b) 描述至少三种方法用于确定滤波器的频率响应函数。
    (c) 描述至少两种方法用于确定滤波器的类型，IIR 或者 FIR。
12. 在滤波器实现过程中，用于计算的位往往多于模-数转换的位。例如，一个系统采用 12 位 A/D，但使用 16 位整型运算以易于处理 32 位的乘积。为什么普遍采用这种方式？计算所利用的额外位是否有实际意义？
13. 如果传输函数的分母为奇数阶，则至少存在一个实数极点。因而可以利用二阶子系统级联结构实现，其中一个子系统必须为一阶或者三阶。
    (a) 画出三阶正则滤波器的框图及其对应的信号流图。
    (b) 求解该三阶正则滤波器的 $\boldsymbol{A}$、$\boldsymbol{B}$、$\boldsymbol{C}$ 和 $\boldsymbol{D}$ 矩阵。
    (c) 说明利用三阶滤波器代替二阶滤波器加一阶滤波器的可能的优缺点。
14. 二阶多项式 z^2+bz+c 的根为 $z=(-b\pm \mathrm{j}\sqrt{b^2-4c})/2$，利用该公式解释 b 的截断常将根移向原点，而 c 的截断将根远离原点(某些情况下)的原因。解释系数的截断如何引起滤波器的不稳定？
15. 解释如何在信号中叠加噪声以改进 A/D 的性能。
16. FIR 滤波器是否可能出现极限环振荡？解释原因。
17. 对于 IIR 滤波器，二阶子系统级联打破了长反馈环，减小了量化误差的影响。那么对于 FIR 滤波器，级联有什么优点？
18. L_∞ 范数缩放方法保证了系统频率响应不出现溢出，其通过增加增益因子限制频率响应的最大值实现。为什么这种方法不能保证系统对于任何输入(该系统可表示的范围)都不出现溢出现象？
19. 本章讨论的三种缩放方法中，L_1 范数缩放的要求最为严格，为什么该方法不适用于所有系统？
20. 在数字滤波器的输出中，乘积量化误差和系数量化误差如何体现？
21. 假设使用 4 位二进制补码系统表示 $-8\sim+7$ 之间的数字，如果改用 8 位二进制补码系统，正数和负数的表示范围是多少？
22. 在浮点数表示的讨论中，有句话是关于尾数的说明的，“由于最高有效位始终为 1，因此不需要对它进行记录，将省略该位。”其含义是什么？为什么这么做？
23. 对于图 8.6 中的 FIR 直接型结构，它包含了 $N+1$ 个乘法器和 N 个加法器，对于图 8.8 中对称结构，如果 N 为偶数需要多少个乘法器和加法器？N 为奇数呢？
24. 对称 FIR 滤波器一定具有对称的二阶子系统吗？请解释原因。
25. 假设 6 阶 FIR 滤波器的多相实现类似图 8.10，但此处选择 6 级。其与直接型实现有什么不同？这么做有什么优点？
26. IIR 滤波器的正则直接 2 型实现所需的乘法器和加法器是否少于非正则直接 1 型实现？请解释原因。
27. 对于给定的 IIR 滤波器，其零极点数量与所有二阶级联的子系统的零极点数量是否相同？如果二阶子系统为并联形式，零极点数量是否相同？
28. IIR 滤波器采用级联或者并联二阶子系统实现时，为什么量化噪声要小于直接使用差分方程实现？
29. 二阶差分方程需要 5 次乘法和 4 次加法运算，假设所有系数不为 0、1 或者 -1，若以状态变量法实现该二阶系统，需要多少乘法和加法运算？
30. 图 8.24 为 $z^2+a_1z+a_2=0$ 的根分布图，其中系数 a_1、a_2 采用 6 位量化。请给出数学证明，解释根位于 0°附件小角度和 180°附近大角度时相对稀疏。
31. 假设有一个 IIR 滤波器为 $H(z)=N(z)/D(z)$。能否通过求解两个 FIR 滤波器 $H_{\mathrm{FIR}1}(z)=N(z)$、$H_{\mathrm{FIR}2}(z)=D(z)$的幅度和相位，求解出该 IIR 滤波器的幅度和相位？利用该方案，IIR 滤波器的幅度可通过 $|H_{\mathrm{FIR}1}(z)|/|H_{\mathrm{FIR}2}(z)|$ 的逐项相除得到，相位可通过$\angle H_{\mathrm{FIR}1}(z)-\angle H_{\mathrm{FIR}2}(z)$得到。
32. 假设 4 阶 IIR 滤波器的分子、分母均无重根。
    (a) 可以组成多少种级联二阶子系统的排列方式？对于 6 阶 IIR 滤波器又是多少？推导通解。
    (b) 可以组成多少种并联二阶子系统的排列方式？对于 6 阶 IIR 滤波器又是多少？推导通解。
33. 低通 FIR 滤波器 $H_1(z)$，生成一个新的 FIR 滤波器 $H_2(z)$，其与 $H_1(z)$基本相同，只是 $H_2(z)$零点的实部为 $H_1(z)$零点实部的负数。换句话说，所有零点为关于复平面 z 虚轴镜像。

(a) $H_2(z)$滤波器的频率响应对比原滤波器有何改变？

(b) 利用原滤波器的脉冲响应表示此镜像滤波器的脉冲响应。

(c) 如果该 $H_2(z)$滤波器具有线性相位且系数对称，两个滤波器的实现形式有什么区别？

分析与设计题

8.1 节

8.1 利用 8 位二进制补码系统表示如下数字：−44、−112.64 和 125。顺序执行下列计算并标出溢出出现的地方：(−44)+(−112)+64+125+(−44)。

8.2 图 8.1 给出了利用 4 位二进制补码系统表示的 −8～+7 之间的数字。从图 8.1 中可以看出，$-0.5\leqslant x<+0.5$ 范围内的数表示为 0000，如果利用全正数系统表示数字 $0\leqslant x\leqslant 16$，请重新画出该图。

8.3 假设一个模拟温度传感器输出电压用于指示 −30～100℃ 的温度范围，精度为±0.15℃。请问需要多少位 A/D，才可以充分利用传感器的温度范围和精度？

8.4 许多小规模系统使用单一电源供电，板上 A/D 接收信号范围为 0～3V。信号的直流分量利用模拟前置滤波器滤除，然后人工生成的直流分量(此情况下为 1.5V)再重新加入。如果输入正弦信号范围为 −1.0～+1.2V，那么利用 A/D 转换后接收到的二进制数的范围是多少？这里采用 10 位 A/D 转换器。

8.5 输入端采用 10 位 A/D 将电压范围为 −5～+5V 的信号转换成二进制补码，8 位 D/A 将正的二进制数转换为 0～10V 范围的模拟信号。如果输入为 1V，转换的 10 位数的低 8 位直接输入 D/A，输出电压将是多少？如果输入电压为 −1V，回答相同的问题。

8.2 节

8.6 假设某给定的滤波器指标生成了 12 阶 IIR 椭圆滤波器，由传输函数提供了两种实现，第一种是利用状态变量的正则二阶子系统的标准实现，第二种为格型 IIR 滤波器实现。如果一个处理器需 M 纳秒完成浮点常数与浮点变量的乘积，需要 D 纳秒完成两个浮点变量的相加，请估计对于每种实现所支持的最大抽样率。这里忽略模-数和数-模转换时间，但是增加 10%的程序开销。

8.7 4 阶椭圆滤波器满足以下指标：

滤波器类型	椭圆低通
抽样频率	11025Hz
低阻带	0～1500Hz
通带	1700～2000Hz
高阻带	2600Hz～$f_s/2$
通、阻带波动	0.01

两个二阶子系统为

$$H_1(z)=\frac{0.04195z^2+0.054929z+0.041953}{z^2-0.93520z+0.29606}$$

$$H_2(z)=\frac{z^2-0.09112z+1}{z^2-0.98694z+0.7288}$$

如果该滤波器由级联二阶子系统实现，需要多少个乘法器？其中与 0、1 和 −1 的乘法不统计在内。利用帕克斯-麦克莱伦方法设计一个满足相同指标的 FIR 滤波器，采用对称结构共需要多少个乘法器？

8.8 图 8.15 给出了转置的 IIR 滤波器，验证对于二阶系统，其状态变量表示为式(8.19)的形式。

8.9 某 DSP 处理器执行一次乘法和加法运算(MAC)需要 m 纳秒，模-数和数-模转换共需 a 纳秒，为下一次循环的平移所有变量的操作需 s 纳秒。对于使用直接型实现的 N 阶线性相位 FIR 滤波器，求出该系统最大的奈奎斯特抽样率的表达式，N 为奇数。

8.10 4 阶低通巴特沃斯滤波器的传输函数为

$$H(z)=1.80597\times 10^{-3}\ \frac{z^4+4z^3+6z^2+4z+1}{z^4-2.7692z^3+3.0105z^2-1.4900z+0.28657}$$

求解使用并联实现的部分分式展开式。

8.11 图 P8.11 给出了 8 阶椭圆低通滤波器的零极点图，如果把该滤波器分解为级联的滤波器子系统，试确定合适的零极点对，使得级联实现中溢出的可能性最小。

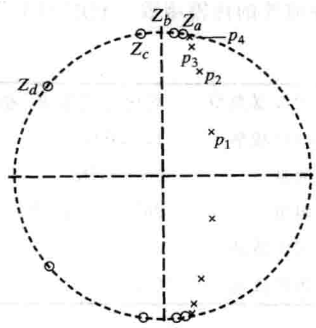


图 P8.11　零极点图

8.12　利用两相结构求出下面低通 FIR 滤波器的多相实现。

$$H(z) = 0.001319\frac{z^7 + 25.088z^6 + 120.51z^5 + 232.28z^4 + 232.28z^3 + 120.51z^2 + 25.088z + 1}{z^7}$$

8.13　利用帕克斯-麦克莱伦方法设计一个满足如下指标的 FIR 低通滤波器：

抽样频率	8kHz
通带	0～1kHz
通带波动	0.03
阻带	2kHz～$f_s/2$
阻带波动	0.03

标出滤波器系数。利用三相结构画出该滤波器的多相实现。

8.14　6 阶 IIR 格型滤波器由下面的 MATLAB 代码中的系数定义，写出仿真程序，利用这些系数和正弦输入产生滤波器的输出，并在同一张图上画出输入和输出信号。改变输入信号的频率以验证该滤波器为低通滤波器。利用式(8.8)和式(8.9)仿真实现每个滤波器子系统。

```
%6th order lattice filter
fs = 11025;T = 1/fs;
fsig = 1000;Tsig = 1/fsig;
t = 0:T:20*Tsig;
%N = 6; %6th order
m = [-0.38279673 0.96366120 -0.60047076 …
     0.69268479 -0.39917519 0.13085309];
c = [-0.03246456 -0.00410914 0.10487847 0.26970878 …
     0.27826164 0.15146278 0.04660996];
x = sin(2*pi*fsig*t);
w = zeros(1, 7);
```

8.15　利用帕克斯-麦克莱伦方法设计一个满足如下指标的 FIR 低通滤波器：

抽样频率	22050Hz
通带	0～3400Hz
通带波动	0.01
阻带	5600Hz～$f_s/2$
阻带波动	0.015

利用 tf2sos 函数将设计的滤波器转换成二阶子系统。写出利用二阶子系统级联实现滤波器方程的 C 代码。通过一个内部循环来求解子系统，外部循环为每个子系统调用内部循环。使用二维数组保存系数，其中第一维为子系统序号，第二维为子系统的系数序号。

8.16 下面是4阶切比雪夫I型低通滤波器的传输函数，利用MATLAB将其转换成格型结构实现。该滤波器满足如下设计指标：

滤波器类型	切比雪夫低通I型
抽样频率	11025Hz
通带	0～900Hz
阻带	2600Hz～$f_s/2$
通带波动	0.005
阻带波动	0.01

$$H(z)=\frac{0.003190z^4+0.01276z^3+0.91914z^2+0.01276z+0.003190}{z^4-2.7085z^3+3.0237z^2-1.6001z+0.3356}=\frac{N(z)}{D(z)}$$

生成频率为f_{sig}的正弦信号x，且为单位幅度，至少包含30个周期。以信号x作为格型IIR滤波器的MATLAB仿真函数latcfilt的输入，并在同一张图上画出输入和输出信号。允许f_{sig}为通带和阻带的边界频率，确定增益是否满足指标。

8.3 节

8.17 画出一个类似给定的表格，说明每个数字如何量化为指定位。每种情况下，选择最小值为0，最大值为2的下一个最高次幂。

	3位	4位	6位	12位
0.5				
2.625				
15.015625				

8.18 下面给出的是某4阶椭圆低通滤波器的传输函数，抽样频率为44.1kHz，利用MATLAB绘出该滤波器的幅频响应曲线，计算1024个点。若将系数量化为b位，其中b为最小的位数，使得量化幅频响应曲线(对于所有的1024点)和原幅频响应曲线之差的绝对值小于或者等于0.01。

$$H(z)=\frac{0.0731534z^4+0.1604528z^3+0.2143748z^2+0.1604528z+0.0731534}{z^4-1.120229z^3+1.1979345z^2-0.5246565z+0.1354232}$$

8.19 某三阶巴特沃斯低通滤波器，抽样率为44.1kHz，其传输函数为

$$H_B(z)=\frac{0.01532z^3+0.04596z^2+0.04596z+0.01532}{z^3-1.8367z^2+1.26166z-0.30237}$$
$$=\frac{0.2326(z+1)}{z-0.53477}\cdot\frac{0.06587z^2+0.1317z+0.06587}{z^2-1.3019z+0.5654}$$

如果以88.2kHz的抽样率设计相同的滤波器，传输函数变为

$$H_B(z)=\frac{0.003025z^3+0.009076z^2+0.009076z+0.003025}{z^3-2.3642z^2+1.91518z-0.52673}$$
$$=\frac{0.13862(z+1)}{z-0.72274}\cdot\frac{0.2182z^2+0.04364z+0.02182}{z^2-1.6415z+0.72880}$$

(a) 你认为两个滤波器对系数的敏感性有什么变化?

(b) 对于每种带有实数极点情况，计算分母中z^2系数的系数敏感性。

8.4 节

8.20 求解下面给出的二阶系统的缩放值，这里分别利用L_1、L_2和L_∞范数方法。使用这些缩放因子在同一个图上画出三个缩放的滤波器频率响应曲线，以及未经缩放滤波器的频率响应曲线，其中$f_s=2$Hz。

$$H(z)=\frac{0.1311z^2+0.2622z+0.1311}{z^2-0.7478+0.2722}$$

8.21 下列6阶椭圆滤波器传输函数将被分解为级联的二阶子系统，其中每个二阶子系统的传输函数形式为

$$H_i(z) = \frac{z^2 + b_{i1}z + b_{i2}}{z^2 + a_{i1}z + a_{i2}}$$

利用 L_2 范数确定三个缩放常数 K_1、K_2 和 K_3

$$H_e(z) = \frac{z^6 + 0.87024z^5 + 2.31066z^4 + 1.77706z^3 + 2.31066z^2 + 0.87024z + 1}{z^6 - 2.32029z^5 + 3.57589z^4 - 3.25146z^3 + 2.04358z^2 - 0.75927z + 0.14353}$$

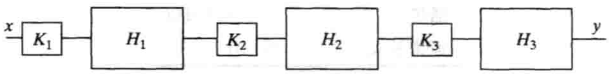


图 P8.21　级联滤波器方框图

8.22　利用 L_1、L_2 和 L_∞ 范数方法求解下面给出的 4 阶系统的缩放值，使用这些缩放因子在同一个图上画出三个缩放的滤波器频率响应曲线，以及未经缩放滤波器的频率响应曲线，其中 $f_s = 2\text{Hz}$。

$$H(z) = \frac{z^4 + 1.2183z^3 + 1.8808z^2 + 1.2183z + 1}{z^4 - 1.9221z^3 + 1.9478z^2 - 0.9737z + 0.2158}$$

8.23　应用式(8.24)～式(8.27)于任意二阶系统，证明当 $\boldsymbol{T} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$ 时，变换后的系统与原系统相同，状态变量的名称在顺序上相反。

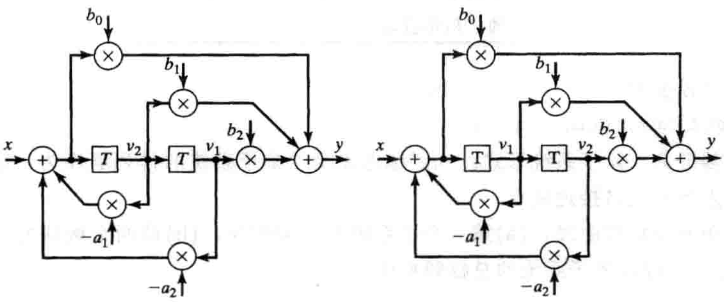


图 P8.23　原滤波器与变换后的滤波器

原　系　统	状态变量交换
$v_1(k+1) = v_2(k)$	$v_1(k+1) = -a_1v_1(k) - a_2v_2(k) + x$
$v_2(k+1) = -a_2v_1(k) - a_1v_2(k) + x$	$v_2(k+1) = v_1(k)$
$y(k) = (b_2 - b_0a_2)v_1(k) + (b_1 - b_0a_1)v_2(k) + x$	$y(k) = (b_1 - b_0a_1)v_1(k) + (b_2 - b_0a_2)v_2(k) + x$
$\boldsymbol{A} = \begin{bmatrix} 0 & 1 \\ -a_2 & -a_1 \end{bmatrix}$, $\boldsymbol{B} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$	$\boldsymbol{A} = \begin{bmatrix} -a_1 & -a_2 \\ 1 & 0 \end{bmatrix}$, $\boldsymbol{B} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$
$\boldsymbol{C} = [b_2 - b_0a_2 \quad b_1 - b_0a_1]$, $\boldsymbol{D} = [1]$	$\boldsymbol{C} = [b_1 - b_0a_1 \quad b_2 - b_0a_2]$, $\boldsymbol{D} = [1]$

8.24　将下列传输函数重新以状态变量形式表示，并利用 MATLAB 确定下列二阶子系统具有无溢出系数：

$$H(z) = \frac{z^2 + 2z + 1}{z^2 - 0.61589z + 0.23139}$$

8.25　采用无溢出状态变量形式，将下列 4 阶传输函数表示为两个二阶子系统：

$$H(z) = \frac{z^4 + 4z^3 + 6z^2 + 4z + 1}{z^4 - 1.9334z^3 + 1.9180z^2 - 0.9623z + 0.2086}$$

8.26　图 8.21 给出了一个以格型结构实现的二阶 IIR 滤波器。设定单位延迟的输出为状态变量 v_1 和 v_2，写出描述该滤波器的状态方程和输出方程。

8.27　利用 MATLAB 设计一个满足以下指标的 IIR 滤波器：

滤波器类型	巴特沃斯带通
抽样频率	1000Hz
下阻带	0～120Hz
上阻带	225Hz～$f_s/2$
阻带波动	0.001
通带	160～185Hz
通带波动	0.001

为了显示该滤波器的频率响应，首先利用MATLAB将IIR滤波器分为二阶子系统形式。计算每个子系统的通带和阻带增益，并显示所有子系统增益的乘积结果。

8.28 设计一个满足以下指标的IIR椭圆滤波器：

滤波器类型	椭圆带通
抽样频率	11025Hz
下阻带	0～1500Hz
通带	1700～2000Hz
上阻带	2200Hz～$f_s/2$
通、阻带波动	0.001

利用MATLAB函数

```
sos = tf2sos(num, den, 'up');
```

将滤波器分为二阶子系统形式。'up'标志表示二阶子系统的排列顺序为：第一个子系统的极点最接近原点且零点最接近极点。

绘出三条幅频特性曲线：(a)第一个子系统的幅频响应，(b)前两个级联的子系统的总幅频响应，(c)所有三个级联的子系统的总幅频响应。

8.5 节

8.29 对于下列给出的二阶系统

$$H(z) = \frac{z^2 + 2z + 1}{z^2 - 0.3z + 0.9}$$

(a) 写出状态变量表达式。

(b) 采用二进制补码溢出规则，当起始点为 $v_1 = 0.95$，$v_2 = -0.95$，输入为零时，绘出矩阵 $\boldsymbol{A}$ 的轨迹曲线。

(c) 应用式(8.24)～式(8.27)以及式(8.36)于该系统，构建一个无溢出振荡的新系统。

(d) 画出新系统的零输入轨迹曲线，起始点为 $v_1 = 0.95$，$v_2 = -0.95$。

8.30 下面的MATLAB程序是一个绘制某二阶子系统状态轨迹曲线的例程，该二阶子系统的状态变量限制为−1～1范围。函数twosover为自定义函数，该函数判断函数输入 x 的值是否在−1～+1范围，应用二进制补码溢出规则返回−1～+1之间的一个数值，可以在附录D中找到该函数。

(a) 编写一个类似例程的MATLAB程序为下列二阶系统

$$H(z) = \frac{z^2 + 1.8z + 0.85}{z^2 - 0.9323z + 0.3675}$$

绘制状态轨迹。

证明在一些初始状态下，该二阶子系统将会溢出。

(b) 采用饱和溢出规则函数satover替换twosover函数。

(c) 利用饱和溢出重复(a)，并比较饱和溢出和二进制补码溢出的轨迹。

```
% State trajectory
v1 = .95;v2 = .95;          %Initial state
CLimit = 50;                %Iteration limit for plot
b0 = 1;b1 = -2;b2 = 1;      %Numerator second order section
```

```
a0 = 1;a1 = 1;a2 = .89;  %Denominator second order section
A = [0 1;-a2 -a1];B = [0;1];C = [(b2-a2*b0) (b1-a1*b0)];D = [1];
% State values are limited to -1 to +1 range
figure(1);clf;
axis([-1.2 1.2 -1.2 1.2]);
line([1 -1],[1 1]);line([-1 -1],[1 -1]);  %Draw a box around the range
line([-1 1],[-1 -1]);line([1 1],[-1 1]);
V = [v1;v2];count = 0;
%Loop until limit is reached or trajectory goes to zero
while(((V(1,1) ~= 0) & (V(2,1) ~= 0)) & count < CLimit)
Vnew = A*V;
% Use twos complement overflow
Vnew(1,1) = twosover(Vnew(1,1));
Vnew(2,1) = twosover(Vnew(2,1));
v = [V(1,1) Vnew(1,1)];              %New point is at (x,y)
   y = [V(2,1) Vnew(2,1)];
line(v, y, 'k');      %Draw line to the new point
V = Vnew;                           %Reset state for next iteration
   count = count + 1;
end
```

8.6 节

8.31 图 8.43 为直接型 FIR 滤波器的 C 代码，改写该代码使其应用于图 8.7 中的转置结构。

8.32 图 8.47 给出了由二阶子系统组成的 4 阶滤波器的 C 代码，使用二维数组重新编写计算两个二阶子系统的代码，其中第一维表示子系统的序号，第二维表示各子系统中系数序号。需要定义 $\boldsymbol{A}$、$\boldsymbol{B}$、$\boldsymbol{C}$、$\boldsymbol{D}$、$\boldsymbol{x}$ 和 $\boldsymbol{zx}$ 的二维数组，列出利用二维数组处理该问题的优缺点。

第9章 数字语音信号

语音频带范围一般为20Hz～20kHz，其与人耳听觉范围相接近。原始的语音信号是时间和幅度上均连续的模拟信号。数字语音是近期的发明，可以追溯到19世纪30～40年代，贝尔实验室发明了脉冲编码调制，第二次世界大战期间语音信号的数字量化首次提出[1]。然而，数字语音信号直到19世纪70年代才获得商业应用，这主要是由于抽样频率和量化误差使得语音的数字信号处理受到限制。相比于数字信号处理的优势，这些限制似乎微不足道，其优势包括针对压缩、传输、接收、录音和音频设备量产等方面的改进技术，其中最为重要的优点就是数字语音可以不丢失任何信息的方式重现语音。

本章将深入讨论语音信号的数字信号处理方法，主要讨论语音信号格式、压缩算法、音响效果，以及均衡器。

9.1 语音信号的性质

可认为声音是空气压力以与人耳频率响应相一致的速率的局部变化。人耳对低于20Hz的压力变化没有反应，尽管它们存在于我们日常环境中。同样，人耳无法听到超过20kHz的声音，该频率上界会随着年龄的增大而降低。话筒将声波转变为电信号，扬声器将电信号还原至声波。

声波信号的频率带宽有限意味着抽样率无须超过最高信号频率的两倍，所以40kHz抽样率适用于所有的声音信号。工业标准设置44.1kHz抽样率作为CD音频格式的标准，对应于22.68μs的抽样间隔。除了44.1kHz之外，对于某些声音信号(如通过电话线的信号)，它们的频带限制在3kHz，其抽样率通常为44.1kHz的约数，如22050Hz或者11025Hz。

语音信号的量化级需要足够精细以捕捉到背景噪声(noise floor)以上的所有信号。如果考虑人耳听觉的下限为0dB，那么类似由割草机、凿岩锤或者重型机械发出的高噪声将达到100dB。对于100dB的范围需要16位量化。刻录在音频CD中的音乐信息为16位，以覆盖人耳听觉的范围。

声波以三种形式记录信息：第一种，幅度变化信息，这就是我们感觉到的“响度”。第二种，信号频率分量的编码。人类语言，对每个抽样值仅用1位信息进行回放也依然可懂，原因在于大量信息内容被编码为信号频率。第三种，记录为相移信息，人类利用相位信息可判断声波方向。

9.2 语音文件编码

声音信号转变为电信号后发送至A/D，为了使得信号处理变得更加有效或者鲁棒，需要对信号进行编码。截至目前，已经提出了多种音频信号的编码方案。这里将讨论三种编码方法：(1)脉冲宽度调制(Pulse Width Modulation，PWM)；(2)脉冲幅度调制(Pulse Amplitude Modulation，PAM)；(3)脉冲编码调制(Pulse Code Modulation，PCM)。表9.1给出了它们的定义。

表 9.1 用于语音信号的典型调制方案

编码	定义
PWM——脉冲宽度调制	脉冲以基频发送。每个脉冲的宽度表示信号在该抽样点处的幅度
PAM——脉冲幅度调制	每个脉冲以基频发送。脉冲的幅度表示信号在该抽样点处的幅度
PCM——脉冲编码调制	编码以基频发送(通常为二进制)。编码表示了信号在该抽样点的幅度

9.2.1 脉冲宽度调制

脉冲宽度调制(PWM),许多微控制器上很常见,利用低通模拟滤波器滤除基频及其谐波,用户可简易地实现数-模转换。图 9.1 给出了正弦信号的典型 PWM 信号。PWM 的优势在于仅需极少的硬件即可产生。除了应用在音频信号中之外,PWM 也常常用于控制信号的输出。图 9.1 给出了 PWM 信号,同时叠加了编码的正弦信号。

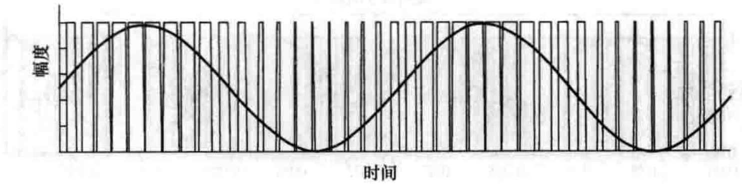


图 9.1 正弦信号的 PWM 编码

在信号的 PWM 中,基频由信号的抽样频率确定。每进行一次抽样会产生一个脉冲,其周期内的脉冲宽度正比于信号幅度。若某信号具有设定最大幅度的一半,其编码为 50% 占空比的脉冲串。编码信号的分辨率取决于编码使用的离散脉冲宽度的数量。对于典型的微控制器中的 8 位计数器,该情况下有 $2^8=256$ 个脉冲宽度。在某些系统中,将 0 信号编码为低电平信号,1 信号编码为高电平信号。8 位系统中信号需要包含时钟,0 信号可编码为 1 个“1”和 255 个“0”,而 1 信号可编码为 255 个“1”跟随 1 个“0”。时钟可由编码信号的上升沿获得。PWM 信号的傅里叶变换显示了原始信号频谱叠加抽样频率及其高次谐波。为了解码 PWM 信号,需要将信号通过低通滤波器以滤除抽样频率以外的分量。

例 9-1 线性调频信号的 PWM 编码

利用 MATLAB 产生一个线性调频信号(chirp),其频率由 0Hz 增加至 600Hz,时间长度在 1s 以上。以 1000Hz 抽样频率、7 位分辨率进行 PWM 编码。显示高频范围编码时的混叠效应。

解: 利用 MATLAB 函数 chirp 生成线性调制信号。为实现 PWM 编码,定义一个数组使每个抽样值占 1 列。本例中,每列有 128 行,如果抽样值为 v,该列填充 v 个“1”和 $128-v$ 个“0”。生成该数组后,将其变换为单列数组并画出对应时间的结果。图 9.2 依次给出了线性调制信号的低频、中频、高频三个部分的信号表示。注意,线性调制信号频率高达 600Hz,高出了奈奎斯特频率 500Hz,导致了混叠。

```
%PWMChirp.m
fs = 1000;T = 1/fs;
N = 1000;
t = 0:T:(N-1)*T;
y = chirp(t, 0, N*T, 600);
b = 7;N2 = 2^b;                    %bits of resolution
y = (y + 1)/2;                     %normalize 0 to 1
%Create the PWM signal in an matrix which has N2 rows x N columns
% where each column represents one sample of the signal.
%Here each column has v ones and N2-v zeros where v is the
% value of the signal at the sample.
```

```
PWMOut = zeros(N2, length(t));
for i=1:size(PWMOut, 2);
    v = fix(y(i)*N2);
    s = [ones(1, v) zeros(1, N2 - v)];
    PWMOut(:, i) = s';
end
%Change PWMOut which is N2 x N to a column matrix that is
% (N2*N) x 1.
PWMPlot = PWMOut(:);
figure(1);clf;
%PWMPlot has N2 times as many samples as t.
Ts = T/N2;
t1 = 0:Ts:N*T-Ts;
plot(t1, PWMPlot);
```

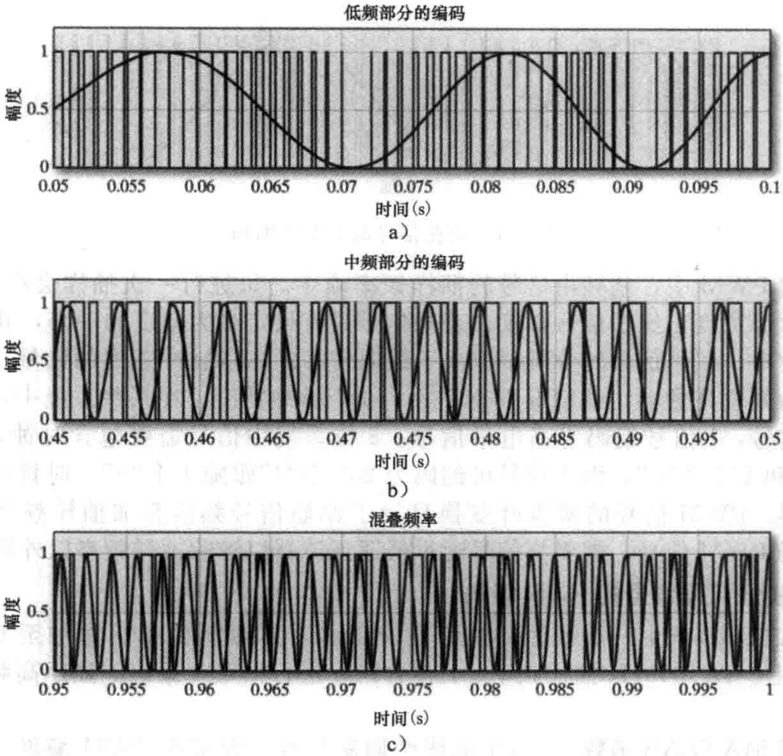


图 9.2 线性调制信号的PWM编码。其中，原始线性调制信号叠加于编码信号。线性调制信号的高频部分超过奈奎斯特频率并存在明显的混叠

9.2.2 脉冲编码调制

脉冲编码调制(PCM)将信号表示为编码序列，每个编码表示抽样点的幅度。可以采用任何形式编码，但二进制编码最为常用。观察包含PCM信号的传输线，可以发现一系列的“1”和“0”以脉冲形式沿着传输线移动。采用脉冲编码调制技术有许多优点。由于发送的信号为二进制数据流，因而抗噪声干扰性能非常好，只要信道不出现位丢失、“1”改变为“0”或者相反情况，信号将在接收端真实地再现。此外，信道的电子特性也不会引起信号质量的下降，信号存储具有相同的情况。PCM信号可以轻松地实现时间复用，使得立体声信号可以在一根线路上实现传输。最后，由于数字编码，PCM可具有相当大的冗余和错误校验，适用于那些数据完整性非常重要的应用。图 9.3 给出了1Hz正弦信号以25Hz

抽样率抽样并编码为PCM信号。利用4位将信号编码为16级，编码后的传输信号为：

{1000，1001，1011，1101，1110，1111，1111，1111，1111，1110，1100，1010，1001，…}

在硬件实现方面，A/D转换器首先将模拟信号转变成PCM信号，然后将其输入典型的DSP处理系统，连同先前的信号进行算法运算，最后将其通过D/A转换器输出。D/A转换器实现将PCM信号转换为脉冲幅度调制信号。

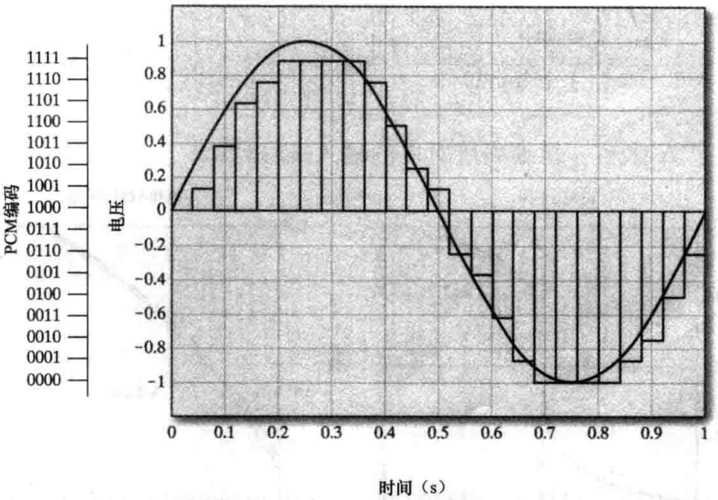


图9.3 PCM编码的正弦信号，4位的PCM编码显示在右侧

脉冲幅度调制(PAM)利用脉冲幅度表示信号抽样点的幅度，其与抽样点具有相对应的幅度值。大多数的D/A转换器内嵌一个输入寄存器，以保存当前值至下一个抽样点的到来，使得输出为阶梯序列而非脉冲序列。将PAM信号通过抗镜像(anti-imaging)模拟低通滤波器滤除$f_s/2$以上频率成分即可恢复原信号。

PCM信号是原始信号的量化形式，可以写为

$$s_{PCM}[n] = s_{analog}[n] + q[n]$$

式中，$q[n]$为量化误差。由4.4节可知均方量化噪声为

$$P_n = \frac{\Delta^2}{12}$$

式中，$\Delta = 2A/2^b$，A为信号幅度，b为位数。

假设信号幅度的峰峰值为1，那么$A=1/2$，这里均方量化噪声变为

$$P_{nDB} = 10\lg[(1/2^{2b})/12] = -6.02b - 10.79\text{dB}$$

作为“经验法则”，PCM信号的量化误差以近似6dB/位速率降低。

PCM有时也称为LPCM，即线性脉冲编码调制(Linear Pulse Code Modulation)。如果我们绘制一个图形，输入信号画在一个轴上，编码输出信号画在另一个轴上，结果将是一条直线。也就是说，输入信号的每个部分与其他部分具有相同的量化级。然而，在某些应用中，这种线性特性并不是所期望的，如语音信号中非常小的信号(窃窃私语)包含与大信号同样多的信息，尽管大信号占用了更多的量化级。由于人耳听觉呈对数特征，所以考虑非均匀编码的对数算法能解决该问题。

目前常用的对数编码有两种方式，即多应用于欧洲的A律编码和多应用于美国与日本的μ律编码，两者非常相似。μ律编码可以表示为

$$y[n]=\frac{y_{\max}\ln\left(1+\frac{\mu|x[n]|}{y_{\max}}\right)}{\ln(1+\mu)}\operatorname{sgn}(x[n])$$

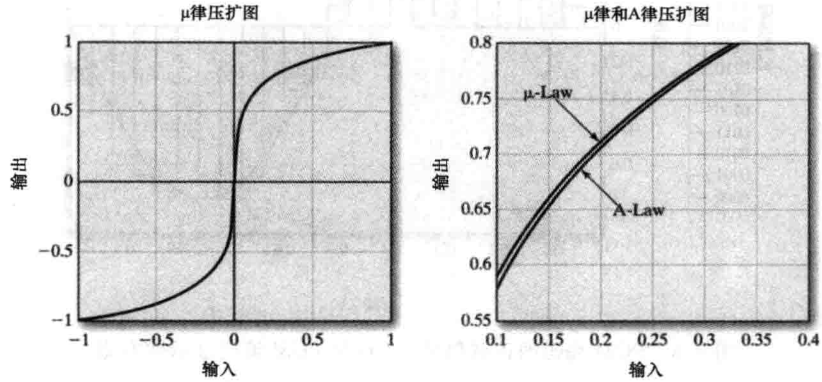
式中，$x[n]$为输入信号，$y[n]$为输出，sgn 为符号函数，μ 值设为 255。

A 律编码可表示为

$$y[n]=\begin{cases}\dfrac{A|x[n]|}{1+\ln(A)}\operatorname{sgn}(x[n]) & 0\leqslant x[n]\leqslant y_{\max}/A\\ \dfrac{y_{\max}\left[1+\ln\left(\dfrac{A|x[n]|}{y_{\max}}\right)\right]}{1+\ln(A)}\operatorname{sgn}(x[n]) & y_{\max}/A\leqslant x[n]\leqslant y_{\max}\end{cases}$$

式中，A 取值为 87.56。

图 9.4 给出了 A 律和 μ 律编码所对应的输入-输出图。



a）μ律编码　　b）局部放大的A律和μ律编码

图 9.4

可以证明，对于每个样本采用 8 位语音信号，A 律或 μ 律编码与采用 12 位线性编码具有相同的信号保真度。表 9.2 给出了范围－1～1 的数字以 4 位 A 律和 μ 律编码的结果。

表 9.2　范围－1～1 的数字以 4 位 A 律和 μ 律编码，数字以二进制补码方式表示

数字	μ律	A律	数字	μ律	A律
−1.0000	−1.0000	−1.0000	0.0000	0.0000	0.0000
−0.8750	−0.9760	−0.9756	0.1250	0.6299	0.6200
−0.7500	−0.9484	−0.9474	0.2500	0.7521	0.7467
−0.6250	−0.9157	−0.9141	0.3750	0.8243	0.8208
−0.5000	−0.8757	−0.8733	0.5000	0.8757	0.8733
−0.3750	−0.8243	−0.8208	0.6250	0.9157	0.9141
−0.2500	−0.7521	−0.7467	0.7500	0.9484	0.9474
−0.1250	−0.6299	−0.6200	0.8750	0.9760	0.9756

9.2.3　微分脉冲编码调制

许多音频信号(如语音和音乐)在相邻的抽样值之间存在某些相关性。因而在编码算法中考虑此因素将更加合乎情理。微分脉冲编码调制(Differential Pulse Code Modulation,

DPCM）对信号的走向可以做出某种预测，然后从信号中减去预测值并对差值进行编码。所以这种编码方法可以使用更少的位实现信号的编码。

考虑输入信号 $x(t)$，计算 $x(t+T)$ 的预测值。如果已知信号 $x(t)$ 的所有微分，其泰勒级数为

$$x(t+T)=x(t)+T\frac{\mathrm{d}x(t)}{\mathrm{d}t}+\frac{T^2}{2!}\frac{\mathrm{d}^2x(t)}{\mathrm{d}t^2}+\frac{T^3}{3!}\frac{\mathrm{d}^3x(t)}{\mathrm{d}t^3}+\cdots$$

如果 T 非常小，可忽略高阶项，上式可写为

$$x(t+T)\approx x(t)+T\frac{\mathrm{d}x(t)}{\mathrm{d}t} \tag{9.1}$$

将式(9.1)表示为差分方程，将导数近似为当前抽样值与前一个抽样值间的斜率。则有

$$x[n+1]=x[n]+T\left(\frac{x[n]-x[n-1]}{T}\right)=2x[n]-x[n-1] \tag{9.2}$$

式(9.2)是 $x[n+1]$ 的一阶预测器。通过 $x[n]$ 的真实值与预测值相减求得差值 $D[n]$，并对差值 $D[n]$ 进行量化和传输，图 9.5 给出了流程框图。量化后的信号等于信号预测值加上传输的差值，可以写为

$$x_q[n]=P_x[n]+D_q[n] \tag{9.3}$$

图 9.5 中的预测器是差分表示式(9.2)的实现，其仅利用了信号的一阶导数。显然，利用高阶导数可获得更好的预测，但预测器表示式的形式都是 FIR 滤波器。

直观来看，DPCM 借助信号相关性的优势，使得利用相同的位数可表示更小的量化级以降低量化误差。利用一阶预测器的差分方程，正弦信号每个周期包含 30 个抽样值，产生的结果如图 9.6 所示。

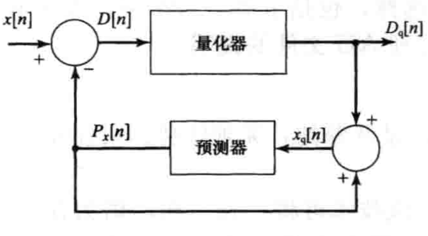


图 9.5　产生 DPCM 的系统框图，输出信号 $D_q[n]$ 等于输入信号减去输入信号的预测值

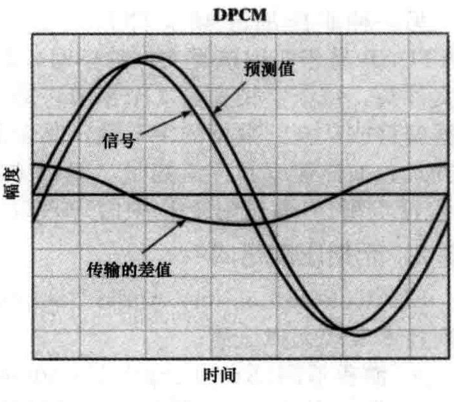


图 9.6　DPCM 的模拟，正弦信号以每个周期 30 个抽样值进行抽样

DPCM 接收机实现差分信号的解码并恢复原始数据，通过图 9.7 所示可以实现式(9.3)表示的系统。

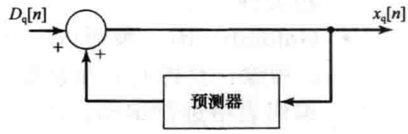


图 9.7　DPCM 解码器

PCM 信号的量化噪声由下式给出

$$P_n=\frac{\Delta^2}{12}$$

其中，Δ 为最小量化级。如果信号的最大幅度为 s_{max}，最大幅度差值为 d_{max}，量化级可以降低 s_{max}/d_{max} 倍，使得量化噪声级降低 $(s_{max}/d_{max})^2$ 倍。此外，传输信号的幅度降低，信号能量与信号幅度平方成正比，所以信噪比可获得更大的改善。

9.3 音频文件格式

音频文件格式指定音频数据存储的编排格式。目前有大量文件格式存在，可以粗略地分为无损和有损、压缩和非压缩类型。非压缩格式都是无损的，原始数据的抽样值连同抽样率和分辨率等信息都保存。典型的非压缩格式有WAV和AU文件格式。压缩文件可以是无损的，在此情况下所有抽样数据都存储，因而可以完整地恢复语音信号。无损压缩常利用音乐或谈话中频繁出现的无声或者抽样值恒定情况来减少文件大小。通常的压缩比例可以达到2∶1。无损压缩文件格式包括FLAC和WavPack格式。有损压缩格式通过降低正常人耳听力之外的声音的分辨率以获得更高倍数的压缩。所以，在压缩过程中造成了信息的丢失，无法完整恢复原始信号。有损压缩技术非常有用，能获得10∶1甚至更高的压缩比例。MP3就是有损音频压缩技术应用的典型实例。

9.3.1 无损文件格式

WAV(Waveform Audio Format)是典型的非压缩格式，最初由Microsoft和IBM公司提出作为个人计算应用的音频文件格式。WAV格式是RIFF(Resource Interchange File Format)格式的子集，以分块的形式指定了多种不同的文件属性。典型的WAV格式文件包含一个主信息块和两个子块，其中主信息块长度为12字节，包含块描述符，用于表明文件为WAV格式，两个子信息块紧随主块之后，第一个子块为24字节，用于指定音频格式、通道数量、抽样速率、字节速率、多通道块校准和每个抽样值位数。第二个子块存储LPCM数据。WAV文件利用C语言读取相对来说非常容易实现，同样，MATLAB函数wavread实现将WAV文件读入数组。(参见附录D中MATLAB函数处理WAV文件的说明。)

另一种非压缩音频文件格式为AU文件类型，由Sun Microsystems公司提出，MATLAB易于实现该类文件的读写，其与μ律编码密切相关。每个AU文件包含6个字的文件头，每个字长度为4个字节，第一个字指示该文件为AU格式文件，第二个字指示数据偏移或从0到数据的字节数。其余的4个字以字节数给出数据大小(字节为单位)、编码方式、抽样速率和声道数量。编码方案有多种选择，包括μ律、LPCM、A律和IEEE 32位浮点型。(参见附录D中的MATLAB函数处理AU文件的说明。)

9.3.2 无损压缩格式

FLAC(Free Lossless Audio Compression)格式是开源的，无须注册即可使用，其采用下列技术实现无损压缩。

- 游程编码(Run Length Encoding，RLE)：该技术可将一长串相同值另存为单个值和一个计数值。例如，长时间的语音静音片段可由静音的编码和静音抽样值个数替代。
- 线性预测(Linear Prediction)：如DPCM一样，线性预测去除了相邻抽样值之间的相关性。
- Golomb－Rice编码(Golomb－Rice Encoding)：待编码的每个字w首先被定点数d相除，获得商q和余数r，由于为整数除法，商和余数均为定点数，可通过查找编码表格进行编码。而q和r的编码长度低于w的编码长度。

所有的FLAC数据皆为整数，浮点抽样值为非法值。FLAC位流由“fLaC”文本字符开始，紧接着至少1个文件信息描述块，然后是包含数据的一个或多个音频帧。所需的元数据块包含块大小、音频数据帧大小、抽样速率、通道数量、抽样值位数、流抽样点数量，以及一些错误校验信息。可以添加任何数量的其他元数据块用于指示其他数据，如图片和查找表。实际中，FLAC可以获得30%～50%的压缩比例。

9.3.3　有损压缩格式

心理声学是研究我们如何感知声音的学科领域。某些音调虽然存在于音频频段，但并不意味着人耳就一定能够听到它。低频的音调可以掩盖高频的音调，幅度同样起到类似作用。有损压缩算法充分利用该现象，通过消除那些文件中被其他声音掩盖的部分以实现压缩。常用的有损压缩算法为 MP3 格式，这是 MPEG-1 Audio Layer 3 的简称，其中 MPEG 表示运动图像专家组(Moving Picture Experts Group)。特别需要指出的是，信息的丢失仅仅发生在 MP3 的编码过程中。一旦语音信号编码之后，信息在传输、记录、接收和回放过程中不再丢失。存储在 CD 中的声音文件通常以 44100Hz 抽样，每个抽样值 16 位，包含双通道，转换为 1411200 位/秒的位速率。MP3 压缩算法可以去除超过 16kHz 以上的大多数无法听到的频率成分，丢掉被其他频率成分掩盖的那些频率成分，以及被高幅度音调掩盖的那些音调。总体结果就是降低所需 bits/s 的数量大约至十分之一。位速率降低至 100～200kbits/s 比较常见。

9.4　音效

我们用于产生数字滤波器的多数技术可经修改应用于音频信号，实现任意信号的产生、延迟、混响、颤音、镶边、合唱以及均衡化。本节中，我们将应用所学到的 DSP 算法用于音效处理，其中多数可以利用 MATLAB 进行仿真。

9.4.1　振荡器和信号发生器

正弦波振荡器常用于音频设备之中，如音乐合成器。这样的振荡器可以利用 DSP 系统进行模拟，使用包括查表方法、多项式拟合、二阶递归振荡器或者复杂的正弦振荡器等多种方法实现。

查表法。查表法正弦波振荡器在存储器中存储正弦值，并以一定速率循环输出这些值，其中输出速率决定振荡器的频率。对于正弦信号，由于四个象限的对称性，正弦表无须存储整个周期的所有值。第一象限的值通过适当地改变其符号即可表示整个周期，而仅须增加少量的计算。通过存储更多的值或者插值方法可以获得更高精度的信号。线性插值最为常见，利用线性插值，如图 9.8 所示，可以得到 $\sin(t_s)$ 的近似

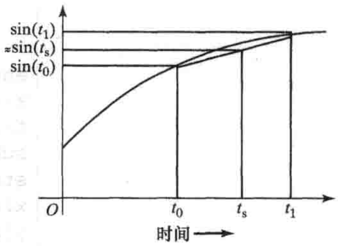


图 9.8　线性插值

$$\sin(t_s) \approx \sin(t_0) + [\sin(t_1) - \sin(t_0)]\frac{t_s - t_0}{t_1 - t_0}$$

对于线性插值，最大误差正比于两点间距离的平方$(t_1 - t_0)^2$。下面的 MATLAB 代码说明如何利用数据表实现线性插值，该数据表包含 91 个输入值表示第一象限的正弦信号。流程如图 9.9 所示。

根据波形表，也可以利用其他形式的内插。由线性插值很容易推广至多项式插值，将线性插值中的直线方程转换为高阶多项式即可实现。同样地，样条插值使用低阶多项式样条拟合小集合的点，使得所有拟合形成一条光滑曲线。对比线性插值，多项式和样条插值均提供更加精确的结果，具有占用更少的存储单元的优势，但需要付出计算复杂度增加的代价。

多项式近似。正弦波振荡器的多项式近似方法不应与多项式插值相混淆，对于多项式近似，可将正弦写为泰勒级数形式

$$\sin(x) = x - x^3/3! + x^5/5! - x^7 7! + \cdots$$

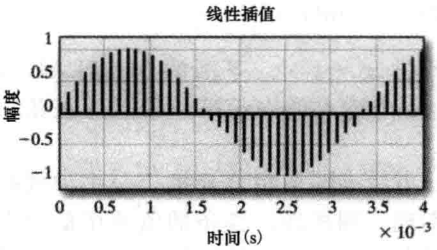
可以写为另一种形式

$$\sin(x) = x(1 - k_1 x^2(1 - k_2 x^2(1 - k_3 x^2 \cdots \tag{9.4}$$

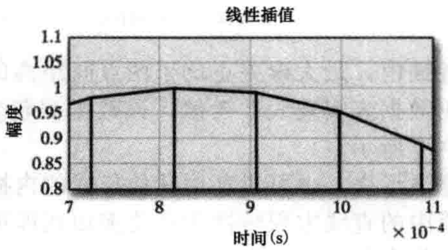
式中，$k_1 = 1/(3 \times 2)$，$k_2 = 1/(5 \times 4)$，$k_3 = 1/(7 \times 6)$，…

```
d = 0:90;
sinTable = sin(d*pi/180);
fs = 11025;
T = 1/fs;
fsig = 300;tsig = 1/fsig;
n = 1;
t = 0:T:.2;
errMax = 0;
y = zeros(1, length(t));
for n = 1:length(t)
    ts = t(n) - tsig*floor(t(n)/tsig);
    index = floor(360*ts/tsig);
    i = floor(index/90);
    switch i
        case 0,
            s1 = sinTable(index+1);
            s2 = sinTable(index+2);
        case 1,
            s1 = sinTable(180-index+1);
            s2 = sinTable(180-index);
        case 2,
            s1 = -sinTable(index-180+1);
            s2 = -sinTable(index-180+2);
        case 3,
            s1 = -sinTable(360-index+1);
            s2 = -sinTable(360-index);
    end
     y(n) = s1 + (360*ts/tsig - ...
                 index)*(s2 - s1);
end
k = 1:length(t);
figure(1);clf;
subplot(2, 1, 1);
stem(t, y, 'MarkerEdgeColor', 'w');
xlabel('time in seconds');
ylabel('amplitude');
title('Linear interpolation');
```

a)



b)



c)

图 9.9　由第一象限数据表利用线性插值方法生成正弦波

由该公式可以看出，x 值越大，收敛所需的时间越长。对于 0～$\pi/2$ 范围内变化的 x 值，级数仅需几项即可收敛。下面的 MATLAB 代码给出了如何利用该级数生成正弦信号。该程序使用级数生成 0～$\pi/2$ 范围的信号值，并利用该第一象限的值获得其他象限 x 对应的信号值。图 9.10 给出了计算结果，式(9.4)由 k_i 确定，$i=0$，1，3。

```
%OscillatorPoly.m
terms = 4; %number of constants
fs = 11025; %sample frequency
T = 1/fs;
t = 0:T:.2; %time variable
fsig = 600; %Signal frequency
for n = 1:length(t)
x = mod(2*pi*fsig*t(n), 2*pi);
  %Determine quadrant and
  % adjust x
  quad = floor(x/(pi/2));
  switch quad
      case 0, %First quadrant
          %x is not altered
      case 1, %Quadrant 2
          x = pi - x;
      case 2, %Quadrant 3
          x = x - pi;
      case 3, %Quadrant 4
          x = 2*pi - x;
  end
  %Accumulate series terms
  s = 1;
  for i=terms:-1:1
      k = 1/(2*i*(2*i+1));
      s = 1 - k*x*x*s;
  end
  sig(n) = s*x;
  %Adjust sign for quadrant
  if(quad == 2 || quad == 3)
     sig(n) = -sig(n);
  end
end
figure(1);clf;
plot(t, sig);
```

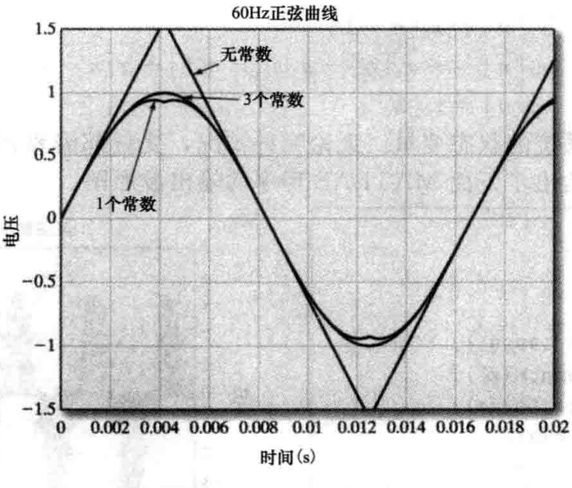


图 9.10　使用截断的泰勒级数生成近似的正弦信号

二阶递归振荡器。递归系统的稳定性需要满足所有极点位于单位圆内。在这种情况下，系统的脉冲响应将最终衰减至 0。如果极点位于单位圆之外，系统的脉冲响应将无限制地增大。如果极点恰巧落在单位圆上，系统响应将不会增加，也不会衰减至零点，而是产生震荡现象。对于单位圆上有两个极点的二阶系统，传输函数为

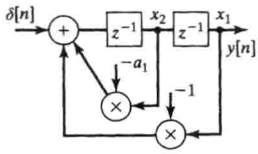
$$H(z)=\frac{1}{z^2-2\alpha z+\alpha^2+\beta^2}$$

其中极点位于 $\alpha\pm \mathrm{j}\beta$，且 $\alpha^2+\beta^2=1$。如果振荡频率为 f_0，抽样频率为 f_s，则 α 可由 $\alpha=\cos(2\pi f_0/f_s)$确定。利用数学推导方式可以得到相同的结果。考虑微分方程 $\frac{\mathrm{d}^2y}{\mathrm{d}t^2}+Ky=x$ 的一个解为 $y=\sin(2\pi ft)$。将一阶、二阶导数近似为两个相邻抽样点间的直线斜率，可以得到

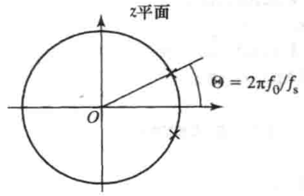
$$\left.\frac{\mathrm{d}y}{\mathrm{d}t}\right|_{t=nT}\approx\frac{y_n-y_{n-1}}{T}$$

以及

$$\left.\frac{\mathrm{d}^2y}{\mathrm{d}t^2}\right|_{t=nT}\approx\frac{\frac{y_{n+1}-y_n}{T}-\frac{y_n-y_{n-1}}{T}}{T}=\frac{y_{n+1}-2y_n+y_{n-1}}{T^2}$$



a）



b）

图 9.11　二阶递归正弦振荡器

将该近似代入微分方程，令 $x=0$ 得 $y_n+a_1y_{n-1}+y_{n-2}=0$，其中 a_1 为常数。此为一个差分方程，其 z 变换为 $z^2+a_1z+1=0$，系统框图和零极点图如图 9.11 所示。

该振荡器也可由下面的差分方程实现

$$y_n=-a_1y_{n-1}-y_{n-2}$$

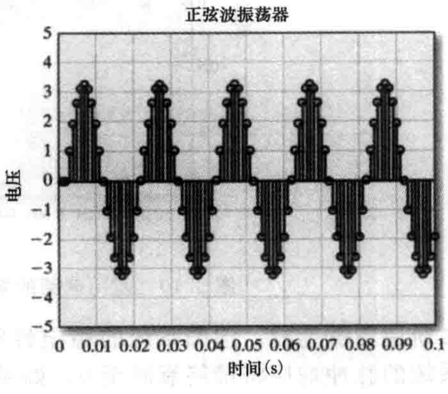
写为状态变量形式

$$\begin{aligned}x_1[n]&=x_2[n-1]\\x_2[n]&=-x_1[n]-a_1x_2[n-1]+x[n-1]\\y[n]&=x_1[n]\end{aligned}$$

这里 x_1 和 x_2 是离散系统的状态变量。无论哪种情况，变量都需要初始化为非零值，以使得系统起振。图 9.2 给出了下面 MATLAB 程序的输出波形图。

```
%Oscillator1.m
fs = 1000;
T = 1/fs;
fo = 50;
x = [1 zeros(1, 19999)];
y = zeros(1, length(x));
alpha = cos(2*pi*fo/fs);
for n = 3:length(x);
    y(n) = x(n-2) +...
        + 2*alpha*y(n-1) - y(n-2);
end
kT = (1:length(y))*T;
stem(kT, y);
```

a）



b）

图 9.12　MATLAB 仿真和结果

三角正弦振荡器。三角正弦振荡器基于两个角度求和的三角恒等式实现

$$\sin(x+y)=\sin(x)\cos(y)+\cos(x)\sin(y)$$
$$\cos(x+y)=\cos(x)\cos(y)-\sin(x)\sin(y)$$

由这两个公式，可实现带有两个输出的振荡器，输出记为 $y_1(n)$ 和 $y_2(n)$，其中 $y_1(n)=\sin(n)$，$y_2(n)=\cos(n)$。我们将参数 n 增大一个微小常量 θ_i，利用三角恒等式可以写为

$$y_1(n+\theta_i)=y_1(n)\cos(\theta_i)+y_2(n)\sin(\theta_i)$$
$$y_2(n+\theta_i)=y_2(n)\cos(\theta_i)-y_1(n)\sin(\theta_i)$$

如果 θ_i 为下一个时间周期的增量，同时令 $\sin(\theta_i)=\mathrm{A}$，$\cos(\theta_i)=\mathrm{B}$，可以得到如下差分方程

$$y_1[n+1]=By_1[n]+Ay_2[n] \tag{9.5}$$
$$y_2[n+1]=-Ay_1[n]+By_2[n] \tag{9.6}$$

例 9-2　三角振荡器

利用式(9.5)和式(9.6)实现输出为 100Hz 的正弦和余弦信号的振荡器，抽样率为 11025Hz。

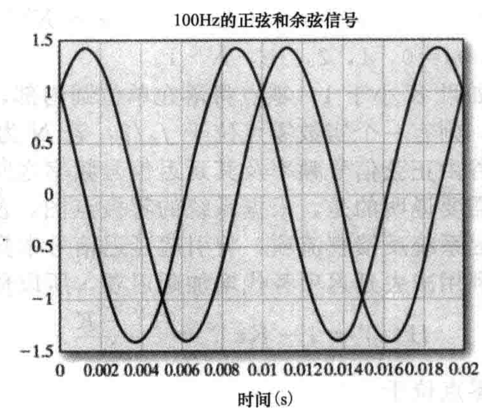
解：信号一个周期对应角度为 2π，θ 的增量由角度确定，该角度对应于一个抽样周期。可以写为

$$\theta_i=\frac{T_{\text{sample}}}{T_{\text{signal}}}2\pi$$

图 9.13 给出了 MATLAB 程序以及生成的波形图。

```
%OscillatorComplex2.m
fs = 11025;Ts = 1/fs;
fsig = 100;Tsig = 1/fsig;
thetaIncr = Ts*2*pi/Tsig;
A = sin(thetaIncr);
B = cos(thetaIncr);
t = 0:Ts:.2;
y1 = [1 zeros(1, length(t)-1)];
y2 = [1 zeros(1, length(t)-1)];
for n = 2:length(y1)
    y1(n) = B*y1(n-1) + A*y2(n-1);
    y2(n) = -A*y1(n-1) + B*y2(n-1);
end;
figure(1);clf;
plot(t, y1);
hold on;
plot(t, y2);
```

a)



b)

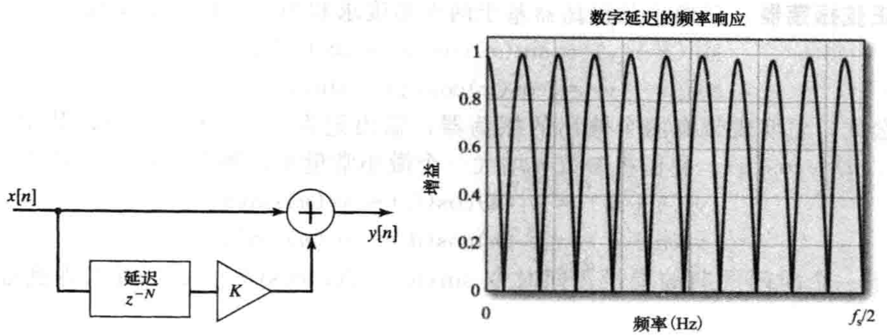
图 9.13　利用式(9.5)和式(9.6)生成的正弦和余弦波形

9.4.2　延迟

时间延迟也许是最简单的音效，仅仅需要一个用于保存延迟抽样的缓冲器即可实现。图 9.14 给出了时间延迟实现框图。传输函数可以写为

$$H_{\mathrm{D}}(z)=1+Kz^{-N}=\frac{z^N+K}{z^N}$$

如果增益常数 K 等于 1，系统就有 N 个零点位于单位圆上，频率响应等同于 FIR 梳状滤波器。



a）时间延迟N个抽样周期的实现框图　　b）对应参数K=1、N=21时系统的频率响应函数

图 9.14　时间延迟

下面的MATLAB程序读取名为“test.wav”的WAV文件，延迟t秒后与原始信号叠加并播放。

```
dtime = .1;
k = 1;
[u fs bits] = wavread('test');
N = fix(fs*dtime);
y = [zeros(N, 1); u(1:length(u)- N)];
y = u + y;
wavplay(y, fs);
```

数字延迟系统零点位于

$$z^N = -K$$

或

$$z = K^{1/N} e^{j\pi(1+2mN)/N}$$

式中，m=0，1，2，…，$N-1$。

如果K小于1，零点将落在单位圆内部，零点之间的频率间隔为$2\pi f_s/N$。如果N为奇数，则有一个实数零点位于$f_s/2$，若N为偶数，则系统不存在实数零点。频率响应的峰和谷由正弦信号频率及其延迟信号频率之间的相长干涉和相消干涉形成。

需要说明的是，根据系统的零极点图，没有零点位于0频率，即$z=1$的情况。原因是延迟系统反馈直流项，将引起延迟信号本身的增加，从而使输出增大。为了改变这种情况，利用减去延迟项替代增加延迟项。所以传输函数变为

$$H(z) = 1 - Kz^{-N} = \frac{z^N - K}{z^N}$$

系统零点位于

$$z = K^{1/N} e^{j\pi(1+2mN)/N}$$

式中，m=0，1，2，…，$N-1$。

零点之间的间隔与延迟增加情况相同，但是，当N为偶数时，存在零点$z=1$和$z=-1$。当N为奇数时，仅有一个实的零点$z=1$。

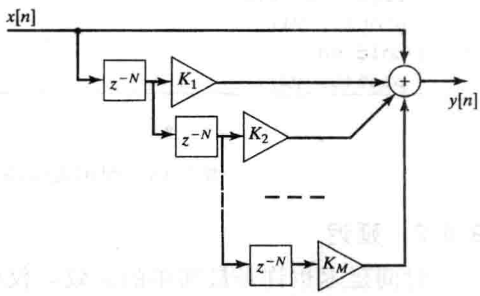


图 9.15　M级时间延迟

假如我们考虑多级延迟系统相加的情形，如图9.15所示，其传输函数为

$$H(z) = \frac{z^{sN} + K_1 z^{(s-1)N} + K_2 z^{(s-2)N} + \cdots + K_M}{z^{sN}}$$

式中，s为级数，N为延迟阶数。

如果 $K_1=K_2=\cdots=K_M=1$，差分方程可写为

$$y[n]=x[n]+x[n-N]+x[n-2N]+\cdots+x[n-sN] \tag{9.7}$$

将此 FIR 函数写为 IIR 方程形式，有

$$y[n-N]=x[n-N]+x[n-2N]+x[n-3N]+\cdots+x[n-sN]+x[n-(s+1)N] \tag{9.8}$$

式(9.7)和式(9.8)相减有

$$y[n]-y[n-N]=x[n]-x[n-(s+1)N]$$

对应的传输函数为

$$H(z)=\frac{z^{(s+1)N}-1}{z^{(s+1)N}-z^{sN}}=\frac{z^{(s+1)N}-1}{z^{sN}(z^N-1)} \tag{9.9}$$

式(9.9)的优势在于分子、分母均可因式分解。因处于原点的极点对幅度函数没有影响，可以忽略它们。其余的 N 个极点位于 $z=e^{j2\pi m/N}$，其中 $m=0$，1，2，…，$N-1$。零点位于 $z=e^{j\frac{2\pi k}{(s+1)N}}$，$k=0$，1，2，…，$(s+1)N-1$。由于系统为 FIR 方程，每个极点必然抵消一个零点，并将发生在

$$\frac{2\pi k}{(s+1)N}=\frac{2\pi m}{N}$$

或者

$$k=(s+1)m$$

例 9-3　求解三级延迟滤波器的传输函数

求解如图 9.15 所示的三级延迟滤波器的传输函数，使用 5 个抽样值延迟，画出结果的频率响应曲线。

解： 令 $s=3$，$N=5$，由式(9.8)可得零点为 $e^{j\frac{2\pi k}{20}}$，其中 $k=0$，1，2，…，19。极点为 $e^{\frac{2\pi m}{5}}$，$m=0$，1，2，…，4。当 $k=4m$ 极点与零点对消，$k=0$，4，8，12，16，剩余的零点位于 $k=1$，2，3，5，6，7，9，10，11，13，14，15，17，18，19，画出结果的零极点图如图 9.16 所示。

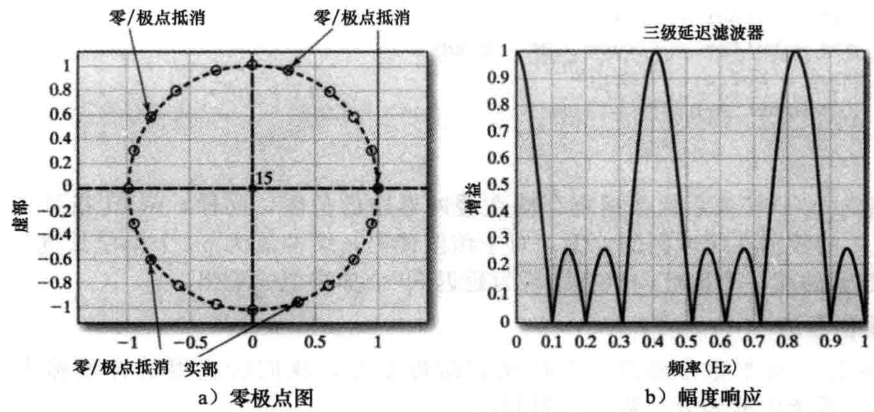


图 9.16　图 9.15 中三级延迟滤波器的零极点图和幅度响应。所有常数均设为 1，延迟为 5 个抽样值

在实时应用方面，数字延迟可使用电路缓冲器高效实现，如图 9.17 所示。数据输入指针指向下一个新数据将存储在缓冲器中的位置，延迟数据输出指针指向将读出的延迟数据位置。每次接收和存储新数据的时刻，两个指针均以缓冲器长度取模值增加。使用这种

方法，数据输入和读出的地址向数据移动的反方向移动，由于缓冲器是环形的，因此旧的数据将被新数据覆盖。

对于某些微型计算机，特别是对于专用的数字信号处理器，环形缓冲器由硬件实现，用户仅需要设置两个指针，指针的增加将由处理器自动完成。如果硬件没有实现环形缓冲器，那么可由软件完成，C代码如下所示。

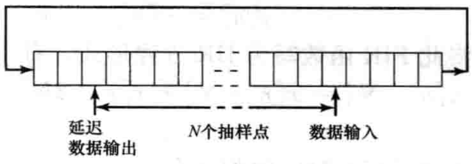


图 9.17　环形缓冲器的实现

```
/*BufferDelayPtrs.c
This program inputs from the AD0 and stores the result in
  a buffer. The buffer output goes to the D/A. The
  buffer is circular. The buffer length delays the input
  and output by N samples. The buffer is managed by
  pointers.
*/
const int N = 50;
int buffer[N];
int *pIn, *pOut;
//
int main(void)
   {unsigned int val, incr;
    //Set up A/D and D/A
    //Start A/D
    incr = 0;
    pOut = buffer + incr; //Initialize pointers
    pIn = buffer + N + incr - 1;
    val = 0;
    while(1)
       {//Read A/D to val
        *pIn = val;
        //Send D/A data from buffer
        D/A = *pOut
        incr = (incr + 1) % N;
        pIn = buffer + ((incr + N - 1)%N);
        pOut = buffer + incr;
        //Restart A/D
       }
  }
```

pIn指针指向A/D转换后的数据将存储在缓冲器中的位置，同样，pOut指针指向将被读出送往D/A的数据在缓冲区的位置。对于该段代码，缓冲器为50个单位长度，输出指针指向放入数据的前一个地址，执行结果为延迟50个抽样值的输出。

9.4.3 调制镶边

对于图9.14中的数字延迟，若将延迟缓慢变化，我们所获得的音效称为调制镶边(flanging)。系统传输函数与数字延迟相同，只是N不再是常数，而变为时间函数。在调制镶边中，延迟通常约10ms，延迟的变化速率一般介于0.2～2Hz之间。图9.18给出了典型的系统框图。

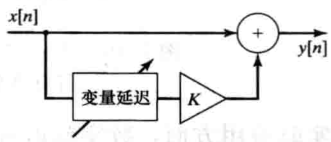
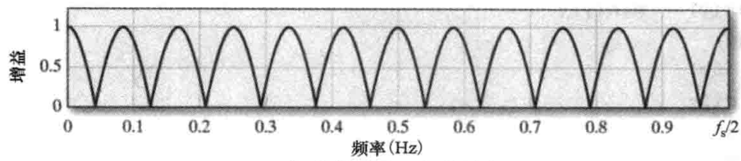


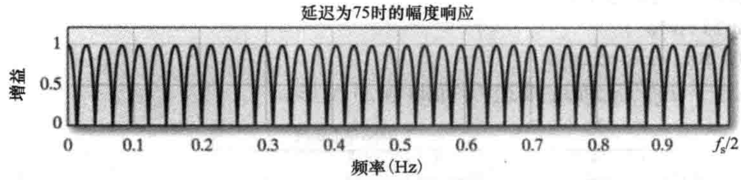
图 9.18　镶边效果的框图，镶边延迟变量为0～10ms，速率变化为0.2～2Hz

为了更好地理解镶边的工作原理，将从频域分析当延迟变量发生改变时的变化。图9.19给出了三种固定延迟的传输函数，延迟从25个变化至

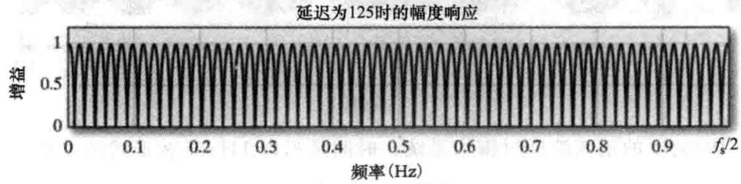
125 个抽样点，当时间延迟增加时，系统零点的个数随之增加，所增强的频带变得更窄。单一的正弦频率通过镶边系统，输出将被交替增强和归零，并生成一个幅度调制信号。



a）信号延迟25个抽样点



b）75个抽样点



c）125个抽样点

图 9.19　数字延迟的幅度响应，分别对应三个不同的延迟量

下面的 MATLAB 代码建立一个镶边系统，时间延迟按照 1Hz 正弦函数由 0 至 10ms 变化，镶边系统的输入信号为 1000Hz 的正弦波，抽样频率 $f_s=11025$Hz。

```
%Flanging.m
%Runs a sinusoid through a flanger and plays the output.
% The delay time is variable from 0 to 10 msec and changes
% as a sinusoidal function.
dtime = .01; %10 msec
f = 1;       %Time variation frequency
K = 1;              %delay gain factor
fs = 11025;
Ts = 1/fs;
t = 0:Ts:Ts*10000; %Total sample length
f1 = 1000;          %signal frequency
u = sin(2*pi*f1*t);
N = fix(fs*dtime);
L = N + 1;          %buffer length
buffer = zeros(1, N+1);
y = zeros(1, length(u));
dIn = N;            %Data in pointer
dOut = 1;           %Data out pointer
for i = 1:length(u)
   buffer(dIn+1) = u(i); %Store a sample
   y(i) = K*buffer(dOut+1) + buffer(dIn+1); %Ouput a sample
   dIn = mod((dIn + 1),L); %Increment input pointer
   %Output pointer is input pointer minus dLen
   % where dLen is variable
   dLen = fix(N*((sin(2*pi*f*Ts*i)+1)/2));
```

```
    %Calculate new output pointer
    dOut = mod((dIn - dLen + L), L);
end
range = max(y) - min(y);
y = y/range;
wavplay(y, fs);
figure(1);clf;
plot(t, y);
axis([.1 .5 -1.2 1.2]);
xlabel('time in seconds');
ylabel('amplitude');
title('1000 Hz sinusoid');
```

图 9.20 给出了正弦波通过镶边系统的输出波形。

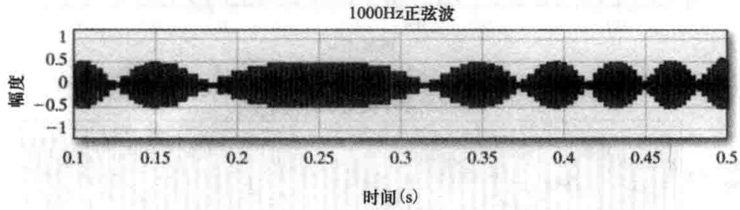


图 9.20 1000Hz 的正弦波通过镶边系统，时间延迟以 1Hz 正弦形式由 0 至 10ms 变化

为实时实现镶边，可以使用空间足够大的环形缓冲器，用于保存最大延迟所需的抽样值。如果缓冲器的长度为 L，最大的延迟抽样点数为 N，且 $L>N$，使用两个指针：第一个指针指向下一个抽样值将被存放的位置，第二个指针指向延迟输出的位置。下面的 C 代码给出了缓冲器指针增加的方法以及镶边输出的计算：

```
/*Flanger.c
This program inputs from the AD0 and stores the result in
 a buffer. The buffer output goes to the D/A. The
 buffer is circular. The buffer length allows a
 sinusoidally varying delay between input and output.
*/
const int N = 50;
const double PI2f = 6.28319;
//
int main(void)
  {unsigned int val, i, K = 1, delayLen, y;
   double Ts = 0.0001; //fs = 10 KHz
   int buffer[N];
   int pIn, pOut;
   //Set up A/D and D/A
   //Set up system clock and timer
   //Start timer
   //Start A/D
   pOut = 0;
   pIn = N - 1;
   val = 0;
   i = 0;
   while(1)
     {//val = A/D input
      buffer[pIn] = val;
      y = K*buffer[pOut] + buffer[pIn];
      pIn = (pIn+1) % N;
```

```
      delayLen = (int)(N*((cos(PI2f*Ts*i)+1)/2));
      i++;
      if(i >= 9999)
        i = 0;
      //D/A = y;
      pOut = (pIn - delayLen + N) % N;
      //Restart A/D Converter
      //Wait for end of sample period
      //Restart timer
      }
  }
```

该程序中，延迟长度以 1Hz 的余弦函数变化，增加了一些注释行以标注某些特定的处理。

9.4.4 合唱

当两个或多个人唱歌或者演奏音乐时，很难长时间完美同步。对这种音效的仿真称为合唱效果或者合唱(chorusing)。为产生合唱音效，需要类似图 9.18 所示镶边系统的电路，但存在两点不同：首先，第一个声音之后的每个声音需要一个可变的延迟电路；另外，可变延迟不应有规律地变化，而应以随机形式缓慢变化。延迟时间长度应在 0～25ms 范围，图 9.21 给出了实时工作方式。

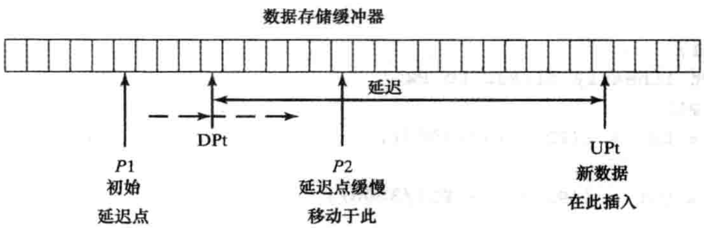


图 9.21　数据存储在缓冲器中，为了获得延迟将位于 UPt 的数据与位于 DPt 的数据相加，对于合唱，*P*1 将慢慢移向 *P*2

我们需要 4 个指针指向存储数据的缓冲器，该缓冲器需要足够大，足以保存最大延迟的数据。进来的数据存于 UPt，Upt 数据与 DPt 数据相加形成合唱音效。DPt 由 *P*1 开始以线性形式慢速移向 *P*2，当 DPt 到达 *P*2 时，令 *P*1 等于 *P*2，并随机选择一个新的 *P*2。每次新数据插入缓冲器，所有 4 个指针均需以缓冲器长度的模值进行移动。

例 9-4　合唱音效的仿真

利用 MATLAB 仿真合唱音效。使用 1000Hz 的正弦波作为输入，抽样频率 f_s = 11025Hz，最大延迟时间为 5ms。

解： 以下 MATLAB 代码实现上述仿真，输入为单一频率的正弦波。

```
%Chorus.m
fs = 11025;T = 1/fs; %sample frequency
fsig = 1000;         %signal frequency
sLen = 20000;        %signal length
dTimeMax = .005;     %5 milli-seconds
K1 = 1;              %gain constant
t = 0:T:T*(sLen-1);
x = sin(2*pi*fsig*t); %input signal
N = fix(fs*dTimeMax);
L = N + 1; %buffer length
buffer = zeros(1, L);
y = zeros(1, length(x));
UPt = N; %Original pointer settings
```

```
P1 = 1;
dLen = fix(N*rand(1,1));
DPt = mod((UPt - dLen), L);
P2 = DPt;
j = 0;
for i = 1:length(x)
   buffer(UPt+1) = x(i); %new input to buffer
   dIndex = mod(round(DPt), L) + 1;
   %Take output from buffer
   y(i) = K1*buffer(dIndex) + buffer(UPt+1);
   UPt = mod((UPt + 1),L); %update all pointers
   P1 = mod((P1 + 1), L);
   P2 = mod((P2 + 1), L);
   DPt = mod((DPt + 1), L);
   %after 3000 samples DPt will equal P1 so reset
   % to a new random value
   if(j == 3000)
     dLen = fix(N*rand(1,1));
     P2 = P1;
     P1 = mod((UPt - dLen), L);
     j = 0;
   end
   j = j + 1;
   %Move DPt linearly closer to P2
   if(P2 > P1)
       DPt = DPt + ((P2 - P1)/3000);
   else
       DPt = DPt + ((P2 + L - P1)/3000);
   end
end
y = y/(max(y) - min(y)); %normalize
wavplay(y, fs);          %play result
```

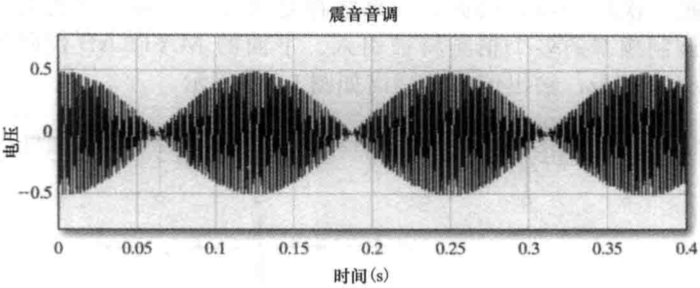
9.4.5 震音和颤音

震音(tremolo)和颤音(vibrato)是两种非常相似的音效，但产生的方式不同。一个无需音乐设备，而另一个需依靠多种设备生成，单独使用声音很难实现。震音是以某一频率改变音符的音量，与幅度调制等效。而颤音则是细微改变一个给定音符的音调，或者是频率调制。两种效果均可由电子方式方便地实现。

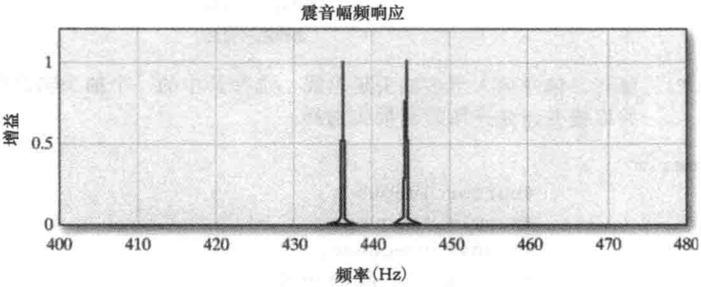
震音可以使用DSP系统实现，将输入信号与一个缓慢变化的正弦波相乘形成信号幅度的调制。下面的MATLAB代码生成440Hz的正弦波作为输入信号，调制信号为4Hz的余弦波。图9.22给出了合成信号的时域波形及其频谱。需要注意的是，由三角恒等式 $2\sin(x)\cos(y)=\sin(x+y)+\sin(x-y)$，在频域中可以得到440Hz±4Hz两个峰值。

```
%Tremolo.m
fs = 11025;T = 1/fs;
fsig = 440;fm = 4;
t = 0:Ts:4;
s = sin(2*pi*fsig*t);
m = .2*cos(2*pi*fm*t);
sm = m.*s;
wavplay(sm, fs);
```

对于颤音，需要调制信号的频率，可由下面的MATLAB代码生成。



a）440Hz的正弦信号由4Hz的正弦信号调制



b）频率响应

图 9.22　合成信号的时域波形及其频谱

```
%Vibrato.m
fs = 11025;Ts = 1/fs;
fsig = 440;Tsig = 1/fsig;
fm = 4;                      %modulating frequency
t = 0:Ts:4;
m = .2*sin(2*pi*fm*t);       %modulating tone
s = sin(2*pi*(fsig+m).*t); %signal tone
wavplay(s, fs);
```

图 9.23 所示为频率调制生成的以信号频率 440Hz 为中心的频率带。

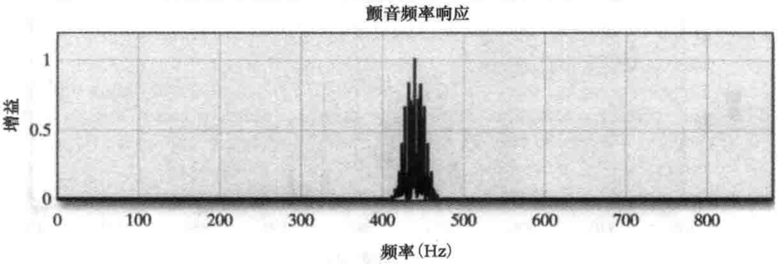


图 9.23　颤音的频率响应，基频为 440Hz

在实时情况下，震音可通过将信号抽样值与慢变正弦抽样值相乘产生，而颤音需要改变信号频率。一种实现方法是将输入信号通过一个类似 FIR 滤波器的单位延迟集合，但选择不同的点作为输出信号，图 9.24 给出了示意图。

例 9-5　利用 MATLAB 仿真颤音的音效

图 9.23 给出了颤音的频率响应，颤音由 440Hz 正弦信号以 4Hz 正弦进行调制形成，抽样率为 11025Hz。写出 MATLAB 代码，利用图 9.24 所示的多抽头延迟线产生相同的音效。

解： 输出信号的频谱由 440Hz 的主音调，加上额外的由调制频率决定的侧音调组成。如果调制频率为 4Hz，则在 444Hz 和 436Hz 处可以得到两个侧音调。在每个调制频率周

期范围内仅需扫描一次缓冲器，因此，如果缓存足够长，边频将获得较少的加强。另外，由于调制步长，调制频率的多个倍频将被引入。下面的MATLAB代码使用长度为10的缓冲器，调制频率为4Hz，输出信号的频谱如图9.25所示。

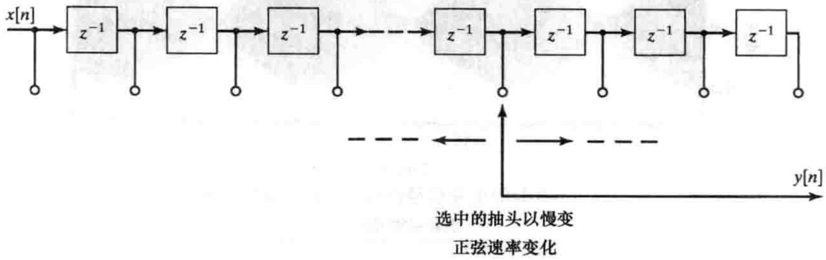


图9.24 颤音。信号输入至多抽头延迟线，选择其中的一个抽头输出信号，并以慢变正弦速率改变抽头选择

```
%VibratoFilter.m
L = 10;                    %buffer length
fs = 11025;                %sample frequency
fsig = 440;                %signal frequency
fm = 4;                    %modulating frequency
t = 0:Ts:4;
m = sin(2*pi*fm*t);        %modulation sinusoid
s = sin(2*pi*fsig*t);      %signal sinusoid
buf = zeros(1, L);
y = zeros(1, length(s));
for i=1:length(t)
    buf(2:L) = buf(1:L-1);  %shift buffer
    buf(1) = s(i);          %new input
    tap = fix(L*(m(i)+1)/2) + 1;
    y(i) = buf(tap);
end
wavplay(y, fs);
```

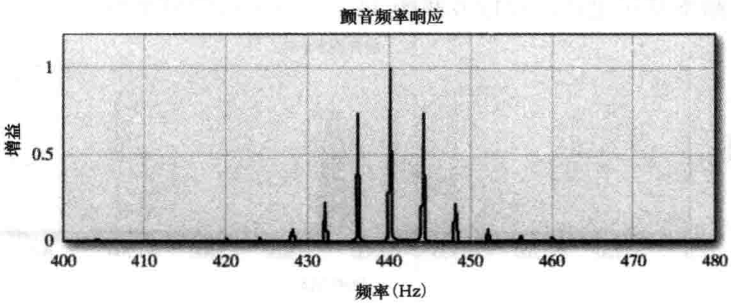


图9.25 使用延迟线滤波器所产生颤音信号的频谱

9.4.6 混响

在密闭空间中，持续时间很短的声音将产生回声，回声本身又将再次产生另一个回声，如此重复。这种重复回声的现象称为混响(reverberation)。人类可以察觉经过大约100ms传回的回声，但我们听到的混响是在更短时间内的许多回声的叠加。

若将混响理解为重复的回声简单易懂，然而要产生合成的或者电子的混响系统却十分复杂。对于音乐厅、大教堂或者简单的房间，都存在许多回声源，每个源都将反射一个不同距离的波，同时声波自身也在反射平面发生改变。

我们首先从最简单的情况着手，构建出一个生成回声降序排列的混响系统的传输函

数。然后通过增加低通滤波器进一步改进该方法，使其更接近现实。也可以将混响看作一个暂态现象，并将全通滤波器作为混响器。最后，将三种基本的混响电路组合，形成一个更丰富、声音更真实的混响器。

首先看一个产生混响的非常简单的例子，图 9.26 中包含两个反射物和一个声源。声源发射声音的脉冲，经过 d 个抽样点之后，反射波出现在声源位置。反射波通过空气时信号有些衰减，并且部分被反射物吸收。反射波继续前行被第二个反射物再次反射，并再次被削弱。该过程将无限地持续下去。

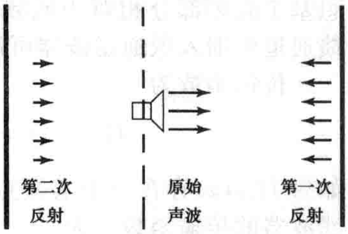


图 9.26　简化的反射模型，原始声波被重复反射

则有

$$y[n]=x[n]+Ax[n-d]+A^2x[n-2d]+\cdots+A^kx[n-kd]+\cdots$$

其中 A 为衰减因子(正实数)。为求该序列之和，有

$$Ay[n-d]=Ax[n-d]+A^2x[n-2d]+\cdots+A^kx[n-kd]+\cdots$$

两式相减求其 z 变换可得

$$Y(z)(1-Az^{-d})=X(z)$$

或者

$$H(z)=\frac{z^d}{z^d-A} \tag{9.10}$$

IIR 系统的脉冲响应为

$$h[n]=\{1,0,\cdots,0,A,0,\cdots,0,A^2,0,\cdots,0,A^3,0,\cdots,0,\cdots\}$$

其中，A 的幂次之间存在 $d-1$ 个零值。

系统所有极点的模为 A^{-d}，相邻极点之间的间隔角度为 $2\pi/d$，并总存在一个极点位于正实轴之上。图 9.27 给出了 $A=0.8$ 和 $d=10$ 时的响应，其幅频响应类似梳状滤波器。

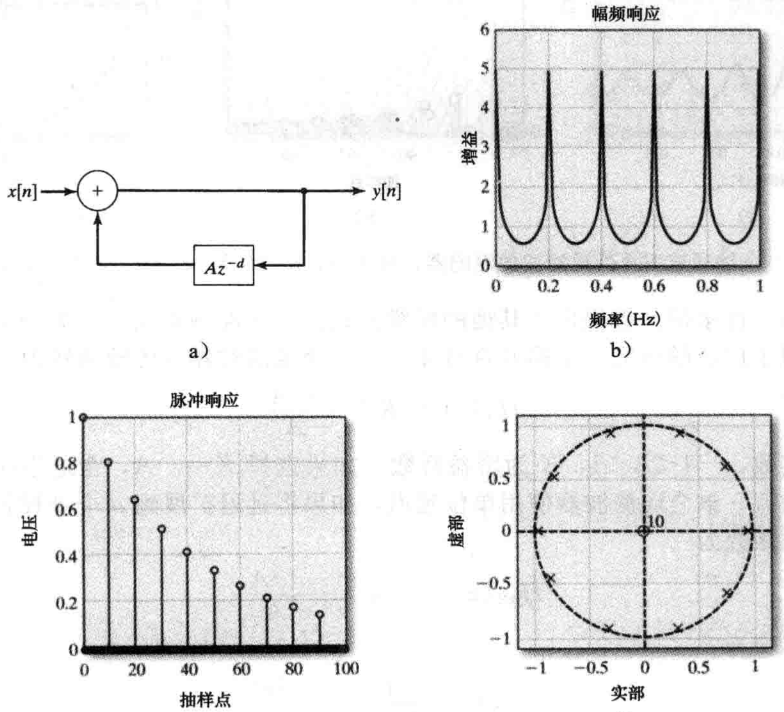


图 9.27　梳状滤波混响器，衰减因子 $A=0.8$，反射间的距离为 $d=10$ 个抽样点

另一种产生混响的方法是在梳状滤波混响器的反馈通道中加入某些滤波模块。这种思想基于高频部分相对于低频部分减弱得更快，所以，通过反馈通道中加入低通滤波器可以实现。框图如图9.28所示。

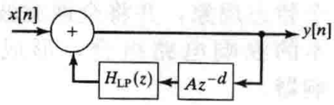


图9.28 反馈通道带有低通滤波器的简化混响系统

传输函数为

$$H_{\mathrm{RLP}}(z)=\frac{z^d}{z^d-AH_{\mathrm{LP}}(z)}$$

如果 $H_{\mathrm{LP}}(z)$ 存在一个实极点 p_1 和一个实零点 z_1，那么低通滤波器的传输函数写为

$$H_{\mathrm{LP}}(z)=\frac{K(z-z_1)}{z-p_1}$$

其中，选取 K 以使得系统在0Hz时的增益为1，即 $K=(1-p_1)/(1-z_1)$。

从而混响器的传输函数为

$$H_{\mathrm{RLP}}(z)=\frac{z^d(z+p_1)}{z^{d+1}-z^dp_1-AKz+AKz_1}$$

对于该类型的混响器，从框图可以清晰看出，低频部分将通过而高频部分将被改变，此通过增强或减弱反馈通道的延迟进行调整。图9.29给出了该系统的频率响应、脉冲响应以及零极点分布图。

对比低通滤波混响器和图9.28中梳状滤波模型的零极点图，可以看出低频极点两者几乎一致，但低通滤波混响器的高频极点相对远离单位圆，因此降低了这些频率的增益。

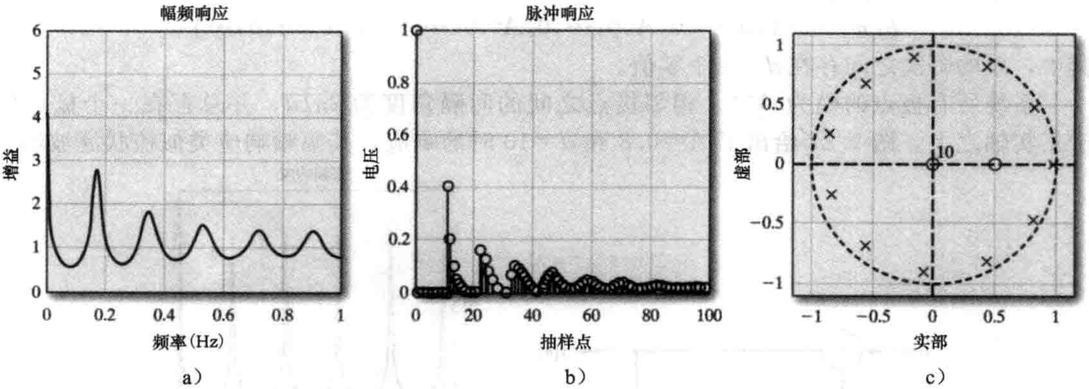


图9.29 反馈通道引入低通滤波的混响器，该例中，$p_1=0.5$，$z_1=0.5$，$A=0.8$，$d=10$

这些年来，许多研究者提出了其他的配置方式。由于混响器是一个暂态效应，因此全通滤波器获得了广泛的研究。由第6章可知，一阶全通滤波器的传输函数为

$$H_A(z)=K\frac{z-1/A}{z-A}$$

其中，A 为实数，$-1<A<1$，K 为增益常数。如果选择 $K=-A$，全通系统对于所有频率都是单位增益。该全通滤波器使用单位延迟，如果将延迟扩展到 d 个抽样值，得到全通混响器的传输函数为

$$H_{\mathrm{AR}}(z)=-A\frac{z^d-1/A}{z^d-A}$$

传输函数也可写为

$$H_{\mathrm{AR}}(z)=\frac{1}{z^d-A}-\frac{Az^d}{z^d-A}$$

该式中，第二部分与式(9.10)中的简单混响器相同。对每项执行长除法可以得到

$$H_{AR}(z) = z^{-d} + Az^{-2d} + A^2 z^{-3d} + \cdots - (A + A^2 z^{-d} + A^3 z^{-2d} + \cdots)$$

系统脉冲响应为

$$h[n] = -A\delta[0] + (1 - A^2)\delta[n-d] + (A - A^3)\delta[n-2d] + (A^2 - A^4)\delta[n-3d] + \cdots$$

所以，全通滤波器的瞬态响应为简单混响器瞬态响应的弱化版本。全通滤波器的优点在于稳态响应是平的，也就意味着当瞬态响应渐渐消失后将不会发生改变。图 9.30 给出了系统的幅频响应、脉冲响应和零极点分布图。

由于这是全通滤波器，在整个频率范围内幅度曲线是恒定的，脉冲响应存在一个初始的负项，其后类似于梳状滤波混响器的柔和形式。图 9.30 中的零极点分布图显示了其与梳状滤波混响器具有相同的极点，而零点则由原点移向了单位圆。

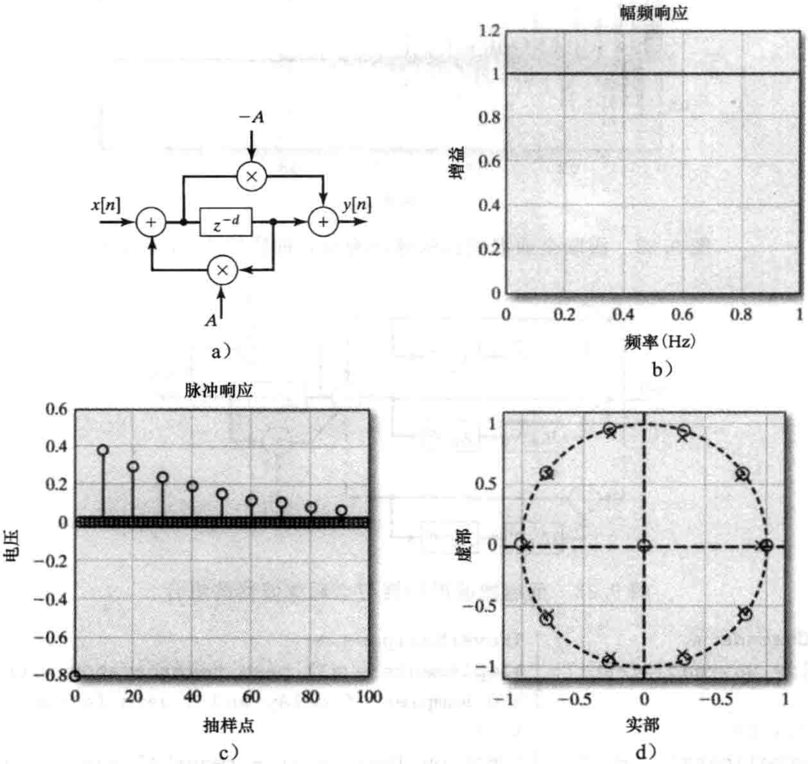


图 9.30　由全通滤波器构成的混响器，其中 $A=0.8$，$d=10$

实际空间中，混响在近似随机的时刻由来自多个源的回声形成，因此最为真实的混响系统可组合利用梳状滤波器、全通混响器和一些低通滤波器，并以不同的延迟时间组成。在这样的系统中，Schroeder 和 Logan[2] 使用级联的全通滤波器，并且每级具有不同的延迟以模拟较为真实的回声。最短延迟时间近似为最长延迟时间的十分之一，三个延迟时间互为质数，以避免系统产生共鸣(共振)。利用 MATLAB 可以方便地模拟一个典型的混响系统，如图 9.31 所示。

下面的 MATLAB 代码读取波形文件，并将信号通过级联全通滤波器，并播放处理结果。级联滤波器系统的脉冲响应如图 9.32 所示。

由一组并联的低通滤波混响器级联一组全通混响器可构成另一种更为复杂的混响结构，此称为 Moore[3] 结构，如图 9.33 所示。

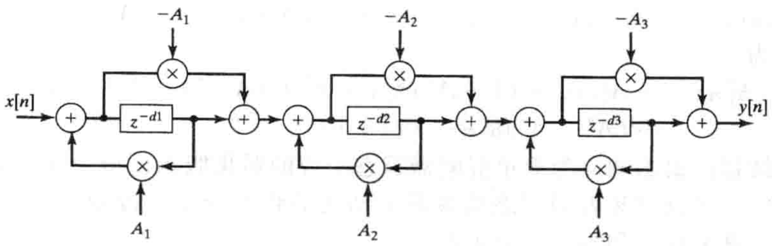


图 9.31　三个全通滤波器级联构成的更接近现实的混响器，时间延迟大致比例为 10∶3∶1

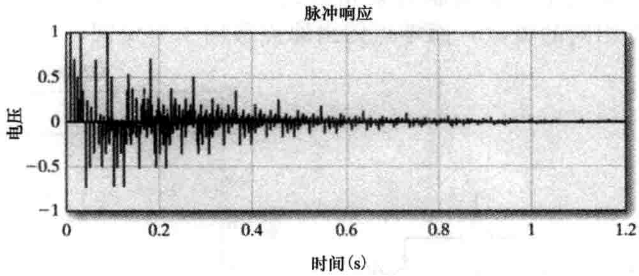


图 9.32　级联全通混响器的脉冲响应，抽样率为 11025Hz

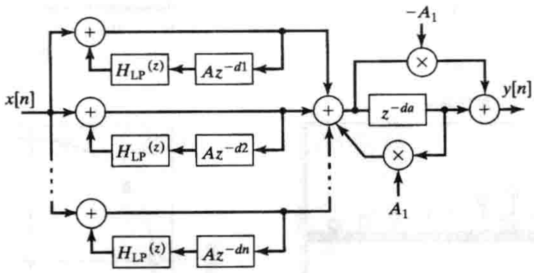


图 9.33　低通滤波混响器与全通滤波器的组合

```
%ReverbCascade.m
[u fs b] = wavread('Test');
A = .7;
d = floor(.091*fs);
y = reverballpass(u, d, A);
d = floor(.033*fs);
y = reverballpass(y, d, A);
d = floor(.0092*fs);
y = reverballpass(y, d, A);
wavplay(y, fs)
```

```
%reverballpass.m
%Implements a all pass reverberator with
% d samples of delay and a gain factor
% of A.
function [yout b a] = reverballpass(u, d, A)
num = -A*[1 zeros(1, d-1) -1/A];
den = [1 zeros(1, d-1) -A];
y = filter(num, den, u);
if(nargout == 1)
    yout = y;
else
    yout = y;
    b = num;
    a = den;
end
```

图 9.33 所示系统的 MATLAB 代码如下所示。例中包含 10 个并联的低通混响器，每个低通混响器具有不同的零极点位置以及不同的延迟时间。抽样率为 11025Hz。

```
%ReverbParallel.m
[u fs b] = wavread('Test');
dAllPass = floor(.006*fs);
```

```
dComb = floor([.008 .009 .012 .031 .0335 .035 .038 .041 .043
.0436]*fs);
p = [.24 .26 .28 .29 .30 .31 .33 .35 .365 .37]; %poles
A = .98;
y = zeros(1, length(u));
for i = 1:10
    y = y + reverblowpass(u, dComb(i), A, p(i), -p(i));
end
A = 0.85;
yout = reverballpass(y, dAllPass, A);
yNm = yout/abs(max(yout));
wavplay(yNm, fs);
%reverblowpass.m
%Implements a all pass reverberator with d samples
% of delay and a gain factor of A.
function [yout b a] = reverblowpass(u, d, A, p1, z1)
K = (1-p1)/(1-z1);
num = A*[1 -p1 zeros(1, d)];
den = [1 -p1 zeros(1, d-2) -A*K*p1 A*K*z1];
y = filter(num, den, u);
if(nargout == 1)
    yout = y;
else
    yout = y;
    b = num;
    a = den;
end
```

Julius Smith 给出了关于基本型和组合型混响器研究方面的最新讨论[4]。

9.4.7 多普勒效应

多普勒效应(Doppler Effect)是一种非常有趣的音效，由运动的声源和静止的收听者产生。如果声源移向收听者，将会听到比传输信号频率更高的声音；如果声源远离收听者，则声音频率会偏低。除了声音频率的改变之外，音量也会随着声源接近变大，远离时变小。当声源接近时频率不断上升，远离时频率不断下降，这通常是一种声音的错觉。其实不然，当声源匀速接近收听者时，听到的声音频率将高于原始频率一个定量，该定量取决于声源移动速度，但听者听到的频率不会发生不断改变，除非声源信号频率改变。当声源远离收听者时会产生相同的音效，只是听到的声音频率将低于传输信号的频率。声源经过收听者时，频率将会发生由高到低的突变。



图 9.34 用于仿真多普勒效应的环形数据缓冲器。数据存放于缓冲器 ptrS 位置，从 ptrD 位置移动数据。指针 ptrS 以缓冲器长度的模增加，且每步的差值是变化的

假设声源频率为 f_{src}，收听者静止，那么听者听到的频率为

$$f = \frac{v_{sound}}{v_{sound} - v_{src}} f_{src}$$

式中，v_{sound} 为声音的传播速度，v_{src} 为声源移动速度，其与声音传播介质相关。

为了模拟该效果，将声音波形存储于包含两个指针的缓冲器中，如图 9.34 所示。

传入的数据流为单一频率 f_{src} 的音频信号，每个抽样值存储在 ptrS 的位置，每个新的数据存储之后 ptrS 以缓冲器长度取模增加。多普勒频移后的音频由 ptrD 取出，ptrD 等于 ptrS－Difference。其中差值在每个抽样周期都有某

微小量的改变。缓冲器的大小需要为最大的差值加1。差值信号改变的速率决定了信号多普勒频移的数量。如果ptrD接近ptrS，则多普勒频率将高于信号频率。如果差值变大，则多普勒频率将低于原始信号频率。

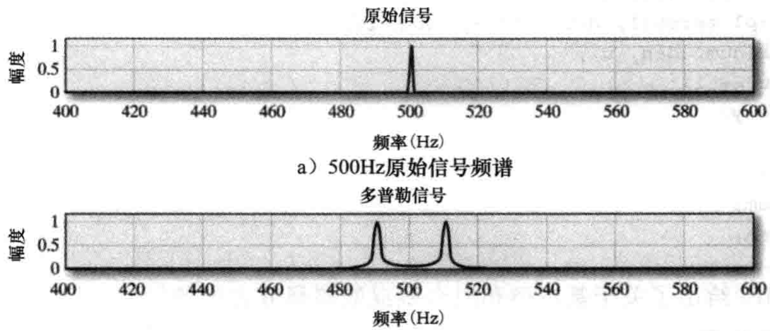
假设在S个抽样值中，我们将ptrD值由缓冲器的最大容量B变化至0，如果传入频率的周期为T_{sig}，则ptrD处产生的新频率的周期为$T_{sig}\pm B\times T_{sig}/S$。

例 9-6 多普勒频移

假设信号的频率为500Hz，如果最大的差值为200，且经过10000个抽样值后差值由最大变换至0，则多普勒频移为多少？抽样频率$f_s=11025$Hz。

解：信号频率为500Hz，则$T_{sig}=2$ms。在10000个抽样值中，周期变化为$\pm(200/10000)\times 2\text{ms}=\pm 0.04\text{ms}$。多普勒频移$f_{D1}=1/2.04\text{ms}=490.2\text{Hz}$，$f_{D2}=1/1.96\text{ms}=510.2\text{Hz}$。

下面的MATLAB代码给出了如何移动环形缓冲器中的数据以生成多普勒频移。本例代码中，声源首先移向收听者，然后远离。代码引入了幅度因子，随着声源移向收听者然后远离，因子由0增加到1，然后再回归到0。原始信号的频谱以及仿真的多普勒信号的频谱如图9.35所示。



a）500Hz原始信号频谱

b）仿真的多普勒频移信号频谱

图 9.35

```
%DopplerBuffer.m
fsig = 500;                        %signal frequency
fs = 11025; T = 1/fs;              %sample frequency
bufSize = 201;
t = T*[1:20000];
s = sin(2*pi*fsig*t);              %signal
buffer = zeros(1, bufSize+1);
ptrS = 201;                        %signal pointer
dif = 200;                         %difference value
ptrD = ptrS - dif;                 %Doppler pointer
%First section. Sound source approaching
for i = 1:10000
    vol = i/10000;                 %volume factor
    buffer(ptrS+1) = s(i);         %signal to buffer
    dp(i) = vol*buffer(ptrD+1);    %Doppler signal
    dif = dif - 200/10000;         %Update pointers
    ptrS = mod(ptrS + 1, bufSize);
    ptrD = mod(floor(ptrS - dif), bufSize);
end
%Second section. Sound source leaving
for i = 10001:20000
    vol = 2 - i/10000;
    buffer(ptrS+1) = s(i);
```

```
        dp(i) = vol*buffer(ptrD+1);
        dif = dif + 200/10000;
        ptrS = mod(ptrS + 1, bufSize);
        ptrD = mod(floor(ptrS - dif), bufSize);
    end
    sound(dp, fs);
```

9.4.8　均衡器

均衡器(equalizer)是用于改变音频信号某一部分频谱而其余部分不变的滤波器。例如，使用均衡器可以提升低音音量，但不改变高频部分音量。均衡器通常由低通和高通倾斜型(shelving)滤波器以及峰值和陷波滤波器(peak and notch filter)级联而成。

一阶倾斜型滤波器。倾斜型滤波器类似于低通或者高通滤波器，但阻带通常设置为 1 而非 0。通带内的频率可增强，称为提升(boost)，或者减弱，称为削减(cut)。低频倾斜型滤波器可提升或者削减低频部分，而高频部分直接通过。高频倾斜型滤波器提升或者削减音频信号的高频端，允许低频部分直接通过。图 9.36 给出了一些低、高频倾斜型滤波器的幅度曲线。

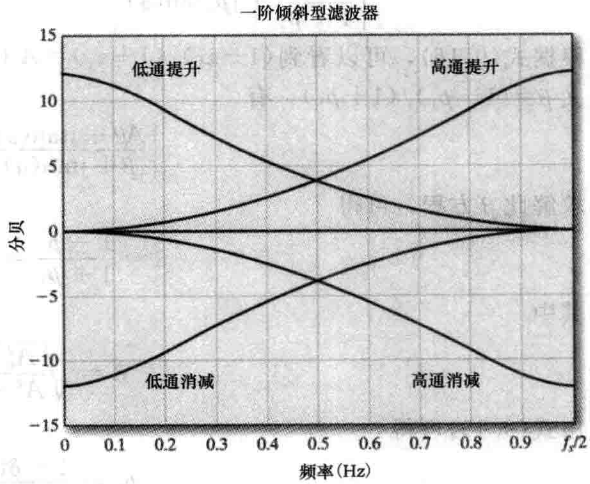


图 9.36　一阶倾斜型滤波器幅度响应

一阶倾斜型滤波器的传输函数为

$$H(z) = K\frac{z - z_1}{z - p_1} \qquad (9.11)$$

图 9.37 中低频提升滤波器的最大增益为 A，频率 f_c 处的增益为 A_c。

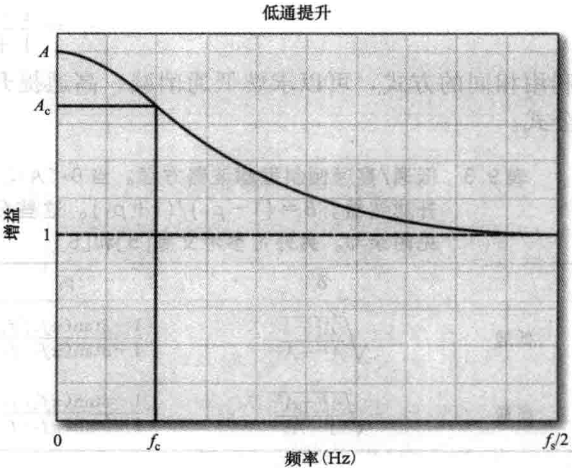


图 9.37　低通提升滤波器，增益 A，截止频率 f_c

由式(9.11)，从图 9.37 中数据点可以写出如下三个方程：

在频率点 0Hz，$z=1$，增益为 A，

$$K\frac{1 - z_1}{1 - p_1} = A \qquad (9.12)$$

在频率点 $f_s/2$，$z=-1$，增益为 1，

$$K\frac{-1 - z_1}{-1 - p_1} = K\frac{1 + z_1}{1 + p_1} = 1 \qquad (9.13)$$

在频率点 f_c，增益为 A_c，z 为一个复数，

$$\left|K\frac{e^{j\theta_c} - z_1}{e^{j\theta_c} - p_1}\right| = A_c \qquad (9.14)$$

求解式(9.13)中的 K 值，并将其代入式(9.12)和式(9.14)，可以获得

$$K = \frac{1 + p_1}{1 + z_1} \qquad (9.15)$$

$$\frac{1+p_1}{1+z_1}\frac{1-z_1}{1-p_1}=A \tag{9.16}$$

$$\left|\frac{1+p_1}{1+z_1}\frac{e^{j\theta_c}+z_1}{e^{j\theta_c}+p_1}\right|=A_c \tag{9.17}$$

由该公式，分子分母提取因子 $e^{j\theta_c/2}$，利用欧拉公式并除以余弦项可得

$$\left|\frac{\dfrac{1-z_1}{1+z_1}+jz_1\tan(\alpha)}{\dfrac{1-p_1}{1+p_1}+jp_1\tan(\alpha)}\right|=A_c，\quad 其中\ \alpha=\theta_c/2=\pi f_c/f_s \tag{9.18}$$

根据式(9.16)，可以看到$(1-z_1)/(1+z_1)=A\ (1-p_1)/(1+p_1)$。将其代入式(9.18)，定义$\beta=(1-p_1)/(1+p_1)$，有

$$\left|\frac{A\beta+j\tan(\alpha)}{\beta+j\tan(\alpha)}\right|=A_c$$

求解此β方程，可得

$$\beta=\frac{1-p_1}{1+p_1}=\delta\tan(\alpha) \tag{9.19}$$

其中

$$\delta=\sqrt{\frac{A_c^2-1}{A^2-A_c^2}}$$

由式(9.19)可得

$$p_1=\frac{1-\delta\tan(\alpha)}{1+\delta\tan(\alpha)} \tag{9.20}$$

最后可得到式(9.16)中 z_1 为

$$z_1=\frac{1-A\beta}{1+A\beta} \tag{9.21}$$

利用相同的方式，可以求解低通削减、高通提升和高通消减滤波器。表 9.3 给出了相关的公式。

表 9.3 低通/高通倾斜型滤波器方程。当 $0<A<1$ 时，生成削减滤波器，当 $A>1$ 时，生成提升滤波器。$\beta=(1-p_1)/(1+p_1)$。这些公式已包含在 MATLAB 函数 equalizer1.m 中，见附录 D。另外可参考文献[5]和[6]

	δ	p_1	z_1	K
低通	$\sqrt{\frac{A_c^2-1}{A^2-A_c^2}}$	$\frac{1-\delta\tan(\pi f_c/f_s)}{1+\delta\tan(\pi f_c/f_s)}$	$\frac{1-A\beta}{1+A\beta}$	$\frac{1+p_1}{1+z_1}$
高通	$\sqrt{\frac{A_c^2-A^2}{1-A_c^2}}$	$\frac{1-\delta\tan(\pi f_c/f_s)}{1+\delta\tan(\pi f_c/f_s)}$	$\frac{A-\beta}{A+\beta}$	$\frac{1-p_1}{1-z_1}$

例 9-7 高通提升滤波器的设计

设计一个高通提升滤波器，峰值增益为 4，抽样频率 $f_s=11025\text{Hz}$，截止频率 $f_c=3,500\text{Hz}$。截止频率处的增益相对峰值衰减 3dB。

解：对于此种情况，$A=4$，$A_c=4\sqrt{2}/2=2.828$。由表 9.3 中的高通滤波器，有

$$\delta=\sqrt{\frac{A_c^2-A^2}{1-A_c^2}}=\sqrt{\frac{2.828^2-4^2}{1-2.828^2}}=1.0694$$

$$p_1=\frac{1-\delta\tan(\pi f_c/f_s)}{1+\delta\tan(\pi f_c/f_s)}=\frac{1-1.0694\tan(3500\pi/11025)}{1+1.0694\tan(3500\pi/11025)}=-0.2469$$

$$\beta=\frac{1-p_1}{1+p_1}=\frac{1+0.2469}{1-0.2469}=1.6557$$

$$z_1=\frac{A-\beta}{A+\beta}=\frac{4-1.6557}{4+1.6557}=0.4145$$

$$K=\frac{1-p_1}{1-z_1}=\frac{1+0.2469}{1-0.4145}=2.1296$$

高通提升滤波器的传输函数为

$$H(z)=2.1296\,\frac{z-0.4145}{z+0.2469}$$

系统的幅频响应如图 9.38 所示。

二阶峰值、陷波和倾斜型滤波器。峰值滤波器类似于共鸣器(谐振器)，只是阻带为 1 而非 0，而且峰值将上升到大于 1 的某个值。图 9.39 给出了一个典型的峰值、陷波滤波器。

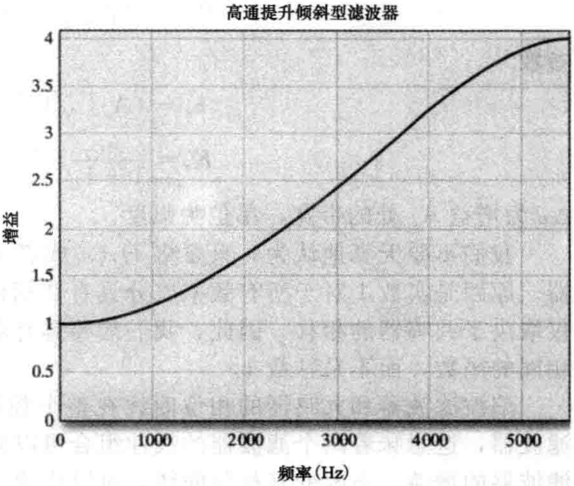


图 9.38 高通提升倾斜型滤波器，峰值增益为 4 ◀

二阶模拟陷波滤波器可以写为

$$H_n(s)=\frac{s^2+\omega_0^2}{s^2+2\delta\omega_0 s+\omega_0^2}$$

为了获得作为均衡器的数字陷波滤波器，仅需要将 ω_0 预畸变后进行双线性变换。经过一些推导，可以得到二阶数字陷波滤波器的传输函数为

$$H_n(z)=K_n\,\frac{a_0z^2+a_1z+a_2}{z^2+b_1z+b_2} \tag{9.22}$$

式中，

$a_0=1$， $a_1=-2\cos(\omega_0)$， $a_2=1$

以及

$$b_1=\frac{-2\cos(\omega_0)}{1+k_1},\quad b_2=\frac{1-k_1}{1+k_1},\quad k_1=(\sqrt{1-A_c^2}/A_c)\tan(\Delta\omega/2)$$

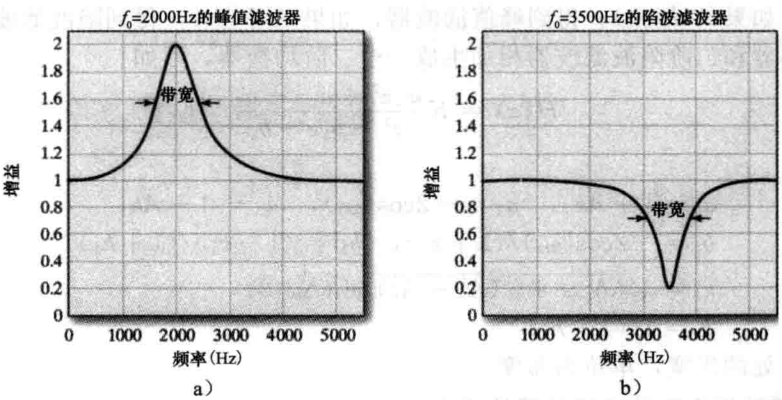


图 9.39 峰值和陷波滤波器作为均衡器。两种情况下，通带宽度在增益为 $\sqrt{(1+A^2)/2}$ 处测量，A 为峰值或者陷波处的增益

且

$$K_n=\frac{1}{1+k_1}$$

$\Delta\omega$ 为增益 A_c 处的带宽，单位为弧度。

对于数字峰值滤波器，同样可由共鸣器作为模拟滤波器推导得到，即

$$H_r(s) = \frac{2\delta\omega_0 s}{s^2 + 2\delta\omega_0 s + \omega_0^2}$$

如果预畸变 ω_0 并利用双线性变换，可以得到 z 域的数字共鸣器，但是基础增益为 0，而不是峰值均衡器所需要的 1。该共鸣器的传输函数为

$$H_r(z) = K_r \frac{a_0 z^2 + a_1 z + a_2}{z^2 + b_1 z + b_2} \tag{9.23}$$

式中，

$$a_0 = 1,\quad a_1 = 0,\quad a_2 = -1$$
$$b_1 = \frac{-2\cos(\omega_0)}{1 + k_2},\quad b_2 = \frac{1 - k_2}{1 + k_2}$$

参数

$$k_2 = (A_c / \sqrt{1 - A_c^2})\tan(\Delta\omega/2)$$
$$K_r = \frac{k_2}{1 + k_2}$$

$\Delta\omega$ 为增益 A_c 处的带宽，单位为弧度。

我们不要天真地认为，只要将 $H_r(s)$ 或者 $H_r(z)$ 加上常数 1，即可得到所需要的滤波器。原因是实数 1 对于所有频率成分具有 0 相移，而此共鸣器不是如此，结果是净增益将仅取决于共鸣器的相移。因此，我们需要具有单位增益，且相移特性与共鸣器的相移特性相同的函数，而不是常数 1。

陷波滤波器和共鸣器的相位曲线在整个频带内相差 90°，这样的一对滤波器称为互补滤波器，这意味着两个滤波器的线性组合可以获得整个频带内的单位增益。改变其中一个滤波器的增益，不改变其相位曲线，可以生成一个新的滤波器，即

$$H(z) = A \cdot H_r(z) + H_n(z)$$

其中，$H_r(z)$ 和 $H_n(z)$ 为共鸣器和陷波滤波器，两者具有相同的带宽指标和相同的 ω_0 值。若远离 ω_0 附近，共鸣器增益为 0，而陷波滤波器增益为 1，所以两者组合后增益为 1。在 ω_0 附近时，共鸣器和陷波滤波器的峰值相加。在 ω_0 时，陷波滤波器增益为 0，使得组合增益等于 A。如果 A 大于 1，得到峰值滤波器；如果 A 小于 1，得到陷波滤波器。

二阶共鸣器和二阶陷波滤波器相加生成一个二阶均衡器，形如

$$H(z) = K \frac{a_0 z^2 + a_1 z + a_2}{z^2 + b_1 z + b_2}$$

式中，

$$a_0 = 1 + Ak_1,\quad a_1 = -2\cos(\omega_0),\quad a_2 = 1 - Ak_1$$
$$b_1 = -2\cos(\omega_0)/(1 + k_1),\quad b_2 = (1 - k_1)/(1 + k_1)$$
$$k_1 = \sqrt{(A_c^2 - 1)/(A^2 - A_c^2)}\tan(\Delta\omega/2)$$
$$K = 1/(1 + k_1)$$

$\Delta\omega$ 为增益 A_c 处的带宽，单位为弧度。

例 9-8 峰值提升和消减滤波器的频率响应

利用 MATLAB 计算多个峰值提升和消减滤波器的频率响应，中心频率为 $f_s/4$，带宽分别为 100Hz、500Hz 和 1000Hz，抽样频率为 $f_s = 11025\text{Hz}$。提升滤波器的增益设置为 1.8，削减滤波器的增益设置为 0.2。

解： 下面的 MATLAB 函数利用了上面峰值滤波器的推导公式，见附录 D。

```
function [num den] = equalizer2(f0, A, bw, fs, Abw)
if(nargin < 4 || nargin > 5)
    disp('error');
    disp('Usage: [num den] = equalizer(f0, A, bw, fs, Abw);');
    disp('[num den] = equalizer(f0, A, bw, fs);')
    return;
end
if(nargin == 4)
    Abw = A - (A-1)/sqrt(2);
end
f0 = f0/fs;w0 = 2*pi*f0;
bw = 2*pi*(bw/fs);
B = sqrt((Abw^2 - 1)/(A^2 - Abw^2))*tan(bw/2);
a0 = (1 + A*B);
a1 = -2*cos(w0);
a2 = 1 - A*B;
b0 = 1 + B;
b1 = -2*cos(w0);
b2 = 1 - B;
num = [a0 a1 a2]/b0;
den = [b0 b1 b2]/b0;
return;
```

调用该函数，设置 A=1.8 生成提升滤波器，A=0.2 生成削减滤波器，结果如图 9.40 所示。◀

例 9-9　二阶峰值滤波器的设计

设计增益为 3 的二阶峰值滤波器，中心频率为 2400Hz，带宽为 800Hz。设置带宽频率为$1+(A-1)/\sqrt{2}$，抽样频率为 11025Hz，所得滤波器的频率响应曲线如图 9.41 所示。

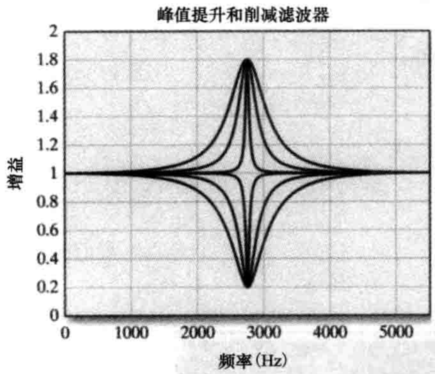


图 9.40　峰值和削减滤波器，带宽分别为 100Hz、500Hz 和 1000Hz

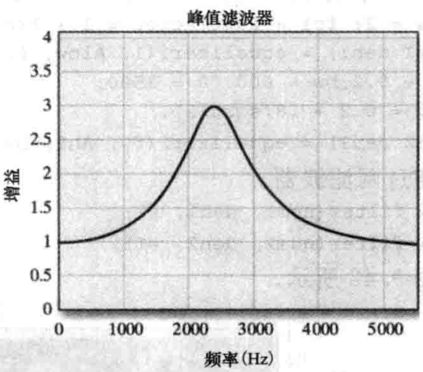


图 9.41　峰值滤波器，中心频率为 2400Hz，带宽 800Hz。该滤波器的 MATLAB 代码已包含在附录 D 的 equalizer2.m 函数中

解：

```
fs = 11025;
f0 = 2400; w0 = 2*pi*f0/fs;
bw = 800; dw = 2*pi*bw/fs;
A = 3;
Ac = (A - 1)/sqrt(2) + 1;
k1 = sqrt((Ac^2 -1)/(A^2 - Ac^2))*tan(dw/2);
a0 = 1 + A*k1;
a1 = -2*cos(w0);
a2 = 1 - A*k1;
b0 = 1;
b1 = -2*cos(w0)/(1+k1);
```

```
b2 = (1 - k1)/(1+k1);
K = 1/(1 + k1);
num = K*[a0 a1 a2];
den = [b0 b1 b2];
[H f] = freqz(num, den, 1024, fs);
figure(1);clf;
plot(f, abs(H));
axis([0 fs/2 0 4.1]);
xlabel('frequency in Hz');
ylabel('gain');
title('Peak Filter');
```

例 9-10 级联均衡器的设计

设计两个均衡器以级联方式处理线性调频(chirp)信号。设置抽样频率为11025Hz，并生成1s的调频信号，其频率范围由100Hz变化至$f_s/2$。第一个均衡器应为低通提升滤波器，其在0Hz处的增益为2，在2000Hz处的增益为$1+1/\sqrt{2}$。第二个均衡器应为高通陷波滤波器，其增益为0.2，中心频率为3500Hz，带宽为600Hz。带宽频率处的增益为$0.2+(1-0.2)/\sqrt{2}$。利用MATLAB中的chirp函数生成信号，利用MATLAB中的filter函数将信号通过两个滤波器，画出输入信号波形，每个滤波器的增益曲线，以及输出信号波形。

解： 下面三行代码生成调频信号：

```
fs = 11025;T = 1/fs;
nT = [0:11024]*T;
s = chirp(nT, 100, 1, fs/2);
```

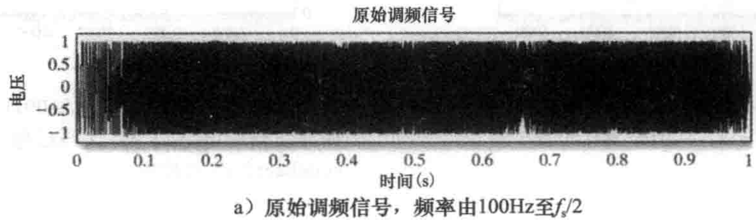
利用附录D中的equalizer1和equalizer2函数生成滤波器。

```
Alow = 2; fc1 = 2000;Aclow = 1 + 1/sqrt(2);
[num1 den1] = equalizer1(1, Alow, fs, fc1, Aclow);
Ahi = 0.2;bw = 600;f0 = 3500;
Achi = 0.2 + .8/sqrt(2);
[num2 den2] = equalizer2(f0, Ahi, bw, fs, Achi);
```

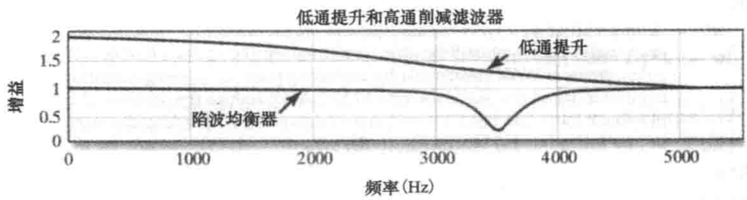
将信号通过两滤波器。

```
s1 = filter(num1, den1, s);
s2 = filter(num2, den2, s1);
```

结果如图9.42所示。

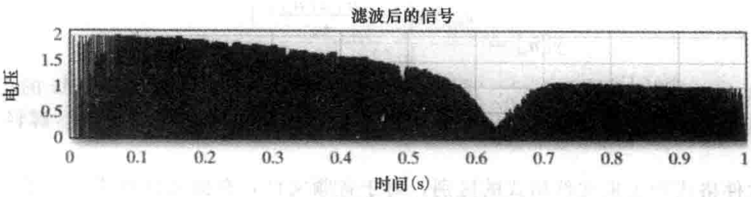


a）原始调频信号，频率由100Hz至$f_s/2$



b）原始调频信号，频率由100Hz至$f_s/2$

图 9.42 例 9-10 图



c）滤波后的信号

图 9.42 （续）

小结

本章介绍了与语音频谱相关的 DSP 知识。由语音文件编码开始，包括 PWM、PCM、PAM 和 DPCM，并介绍了能够获得更好量化效果的 A 律和 μ 律压扩编码。

对于语音音效，给出了 4 种振荡器的设计方法：查表方法、多项式拟合、递归振荡器和三角函数振荡器，并给出了单延迟和多延迟、混响、颤音、镶边、合唱等音效，以及模拟多普勒音效的方法。用于混响的 MATLAB 函数见附录 D。

本章最后给出了均衡器、一阶高通/低通提升和削减滤波器的设计方法，以及二阶峰值/陷波滤波器设计方法，用于均衡器设计的 MATLAB 函数见附录 D。

习题

思考题

1. 图 9.19 所示为不同延迟量的镶边器幅频曲线图，利用此结果解释为什么正弦信号通过不同延迟的镶边器后，得到了图 9.19 所示结果。
2. CD 唱片采用 16 位抽样，而网络上常见的 WAV 格式音频文件为 8 位。请猜测用于记录语音信号所需的最小位数，要求信息不丢失并保证近处的听者能够听懂。例如，如果将从 1 数到 10 的语音信号记录在一个 WAV 文件中，将每个抽样值量化至 3 位，经回放是否还可以分辨所说的数字？猜想之后，利用个人计算机上的声卡生成一个 WAV 格式文件用于记录你的声音，使用 MATLAB 读入该文件，降低每个抽样值的位数，回放该结果直至不能听清为止。若量化位数至 b 位，完成以下工作：(a) 归一化抽样值至 0～1 范围，(b) 乘以 2^b 并截断至整数，(c) 除以 2^b，(d) 去归一化至 −1～1 的范围。
3. 回答以下关于简单混响器的问题，其传输函数为

$$H(z)=\frac{z^d}{z^d-A}$$

   (a) 描述系统脉冲响应和频率响应随 A 值的变化情况；
   (b) 描述系统脉冲响应和频率响应随 d 值的变化情况。
4. 图 9.31 中，全通混响器以级联方式排列，这种排列为什么比并联形式能取得更好的效果？
5. 倾斜型滤波器的作用是什么，为什么利用其建立均衡器，而不利用标准的低通、高通滤波器？
6. 均衡器用于抑制或者提升语音频谱中某些频带分量，以提供听者一个更好的听觉效果，给出至少三个使用均衡器的其他原因。
7. 假设共鸣器(谐振器)在 f_0 处的最大增益为 1，其传输函数为 $H_r(z)$，现在构建一个新的传输函数 $H(z)=1+H_r(z)$，$H(z)$是峰值均衡器吗？说明理由。
8. 低通削减滤波器与高通滤波器相似，同样，高通削减滤波器与低通滤波器相似，请解释它们之间的区别。

分析与设计题

9.1 节

9.1　利用 MATLAB 生成频率为 100Hz 的三角波信号，幅度由 0 至 −1 变化。利用 PWM 编码该信号，抽样频率为 1000Hz，分辨率为 7 位。重叠绘出三角波与 PWM 信号，至少包含 10 个周期的信号。

9.2 节

9.2　μ 律对数编码表示式如下

$$y[n]=\frac{y_{\max}\ln\left(1+\frac{\mu\left|x[n]\right|}{y_{\max}}\right)}{\ln(1+\mu)}\operatorname{sgn}(x[n])$$

利用MATLAB函数实现该表示式，其中$\mu=255$。利用写出的函数编码1Hz的正弦信号，包括4个不同的幅度。在同一坐标轴下绘出原始正弦信号和编码信号，并利用图形解释μ律编码的工作方式。

9.3　解释有损文件格式和无损文件格式的区别，对于音频文件，有损文件格式丢失了什么信息？

9.3节

9.4　图9.8给出了如何进行线性插值，以及利用查表法生成正弦波的方法。

(a) 正弦和余弦之间存在$\cos(\theta)=\sin(90-\theta)$的关系，修改图9.8中的MATLAB代码用于生成正弦和余弦波形，并在相同的坐标轴下绘出波形。

(b) 利用(a)的结果生成$\tan(\theta)$。由于当$\cos(\theta)$趋于0时，$\tan(\theta)$趋于无穷，所以在绘出图形时，需要人工限制该值的大小。

9.5　在9.4.1节中，可利用多项式表示正弦波

$$\sin(x)=x-x^3/3!+x^5/5!-x^7/7!+\cdots$$

该式可以另写为如下形式以更加便于计算，如式(9.4)所示，

$$\sin(x)=x(1-k_1x^2(1-k_2x^2(1-k_3x^2\cdots$$

利用相同做法，余弦波形可以写为

$$\cos(x)=x-x^2/2!+x^4/4!-x^6/6!+\cdots$$

将上述余弦表示式写为类似式(9.4)的形式，并编写MATLAB程序仿真余弦振荡器。

9.6　图9.11为二阶递归振荡器，图9.12给出了差分方程形式的MATLAB仿真实现。利用状态变量重做该仿真。

9.7　图9.12中，MATLAB代码使用输入向量x表示单位脉冲，由于单位脉冲只有在原点处为1，其余均为0，因此可以去除向量x，请重写图9.12没有输入向量$\boldsymbol{x}$的代码。

9.8　三角振荡器的公式如下(来自式(9.5)和式(9.6))，

$$y_1[n+1]=By_1[n]+Ay_2[n]$$

$$y_2[n+1]=-Ay_1[n]+By_2[n]$$

画出该振荡器的原理框图。

9.9　图9.13中的三角振荡器的信号频率为100Hz，抽样频率为11025Hz，求解与100Hz频率相混叠的最低信号频率[⊖]。重新编写对应该频率的程序代码，利用MATLAB验证求解结果。

9.10　设计一个三角波振荡器，要求振荡频率为440Hz，抽样频率为22025Hz，输出幅度为1的正弦波和余弦波。

9.11　根据2011年的吉尼斯纪录，有史以来最低的口哨音符是C调下的F音，由加拿大的Jennifer于2006年创造，该音符对应频率为174.614Hz。同样，Jennifer保持着最高的口哨音符纪录，为C调上的第三个E音，该音符对应频率为1318.51Hz。设计二阶递归振荡器用于生成最低频率音符，设计一个三角振荡器用于生成最高频率音符，并利用MATLAB中的wavplay函数播放。抽样频率为8000Hz，幅度为1V。

9.12　修改图9.13中三角正弦振荡器的MATLAB代码，使得该振荡器产生一个正弦波，要求该正弦波从100Hz开始以2Hz的增量扫频2s以上。为使得频率平滑变化，仅当正弦波由负值变为正值或0值时改变信号频率。

9.13　设计一个二阶递归振荡器，要求振荡频率为1kHz，抽样频率为11025Hz，输出幅度为1的正弦波和余弦波。

9.14　下面的MATLAB代码利用chirp函数生成一个WAV格式的文件，该信号频率在3s的时间内由10Hz变化至1000Hz，抽样频率为8kHz。

```
fs = 8000;T = 1/fs;
t = (0:3*fs)*T; % 3 seconds
u = .49*(chirp(t, 10, 3, 1000)+1);
wavwrite(u, fs, 'ChirpWav');
```

⊖　两者频率不同，但抽样序列的结果相同。——译者注

生成如同图 9.13 所示的单级延迟滤波器，其中 $k=1$。读入 ChirpWav 文件并利用 filter 函数滤波，利用 wavplay 函数回放处理结果。设置延迟时间分别为 0.08s、0.01s 和 0.001s，画出每种情况下滤波器的频率响应以及零极点分布图。作为娱乐，你可以试着确定可以听出原始 WAV 文件和滤波后文件两者区别的最小延迟时间。

9.15 一个信号由方波构成，其幅度变化由 -1 至 1，占空比为 50%，频率为 1kHz，以 20kHz 进行抽样。如果该信号通过增益常数 $k=-1$ 的单级延迟滤波器，分别画出延迟为如下情况的输出信号草图：(a) 2 个抽样点；(b) 10 个抽样点。

9.16 对于 s 级延迟滤波器，每级均为 N 阶延迟，传输函数为

$$H(z)=\frac{z^{sN}+K_1z^{(s-1)N}+K_2z^{(s-2)N}+\cdots+K_M}{z^{sN}}$$

例 9-3 举例说明了该公式在 $N=5$、$s=3$ 时等价的 IIR 形式。利用 MATLAB 证明传输函数 $H(z)=\dfrac{z^{15}+z^{10}+z^5+1}{z^{15}}$ 等价于例中的 IIR 函数。

9.17 利用 MATLAB 生成一个三级延迟滤波器，其中延迟阶数为 5，且所有的增益均为 1，抽样频率为 8kHz。如果幅度变化为 $-1\sim1$、占空比为 50%、频率为 200Hz 的方波通过该滤波器，画出输出信号一个周期内的草图，并利用 MATLAB 验证结果。

9.18 为了更好地理解镶边器如何工作，画出分别对应三种不同延迟量的增益-频率曲线。选择适当的抽样频率，延迟分别为 2ms、10ms 和 20ms。

9.19 利用 MATLAB 仿真合唱效果，选择线性调频信号作为输入，抽样频率为 11025Hz。信号在 3s 时间周期内由 20Hz 变化至 2kHz，最大延迟时间为 5ms。

9.20 如单一频率相同，振音的音效也可以用于一组频率。对于钢琴，C 大调和弦由 C、E 和 G 调组成，分别对应频率 261.626Hz、329.628Hz 和 391.995Hz。生成这三个频率的正弦波并将其相加形成 C 和弦。抽样频率为 11025Hz，归一化信号幅度至 1，利用 MATLAB 中的 wavplay 函数去听结果。生成频率为 4Hz、幅度为 0.2 的余弦波，利用该余弦信号调制和弦信号，如图 9.22 所示。归一化所得信号的幅度为 1，并利用 wavplay 播放结果。利用快速傅里叶变换(FFT)画出原始信号和具有振音信号的频谱。

9.21 将颤音用于习题 9.20 生成的 C 大调和弦，利用频率为 4Hz 的正弦波调制信号频率，如图 9.23 所示。归一化所得信号的幅度为 1，并利用 wavplay 播放结果。利用 FFT 画出原始信号和具有颤音信号的频谱。

9.22 组合低通滤波器与简单混响器的传输函数，传输函数变为

$$H_{\text{RLP}}(z)=\frac{z^d}{z^d-AH_{\text{LP}}(z)}$$

其中，$H_{\text{LP}}(z)$ 为低通滤波器的传输函数

$$H_{\text{LP}}(z)=\frac{K(z+1)^2}{z^2-2c_1z+c_2}$$

(a) 若 $K=0.2066$，$c_1=0.1848$，$c_2=0.1958$，求解传输函数 $H_{\text{RLP}}(z)$。

(b) 若 $A=0.8$，$d=10$，画出 $H_{\text{RLP}}(z)$ 的频率响应。

(c) 若 $A=0.8$，$d=10$，画出 $H_{\text{RLP}}(z)$ 的脉冲响应。

(d) 此应用中使用二阶低通滤波器而不是一阶低通滤波器，其优点和缺点是什么？

9.23 生成一个线性调频信号，起始频率为 100Hz，经 3s 时间到达最终频率 800Hz，抽样频率为 2kHz。构建一个单级全通混响滤波器，其中 $A=0.8$，$d=50$。利用 MATLAB 中的 filter 函数将线性调频信号通过该混响滤波器，播放原始信号和滤波后信号。解释为什么滤波后信号低频部分出现“颤音”，而高频部分出现不含颤音的清晰回声。

9.24 例 9-6 给出了仿真多普勒频移的 MATLAB 代码，修改代码以生成翻转的多普勒频移，也就是时间向后移动，低频信号首先被听到，随后是高频信号。

9.25 多普勒频移常使听者产生一种错觉，即声源逐渐靠近时听到了连续的频率变化，而声源远离时出现相反的变化。认为这是由声源的响度改变所引起。利用 MATLAB 生成真实声音仿真多普勒频移现象，利用 MATLAB 中的 chirp 函数生成线性调频信号，其在 1s 时间内频率由 480Hz 变化至 500Hz，抽样频率设为 11025Hz。然后在下 1s 内频率由 500Hz 降至 480Hz，并以线性方式改变信

号的幅度，使得500Hz时信号响度达到最大。

9.26 利用MATLAB生成低通均衡器的曲线图，其极点位置按步长0.1由0.1增加至0.9，如果(a)该滤波器为低通提升均衡器，在0Hz处的增益为2.0；(b)该滤波器为高通提升均衡器，在$f_s/2$处的增益为2.0。在同一幅图中绘出所有曲线，其中$f_s=2\text{Hz}$，$f_c=0.5\text{Hz}$。

9.27 二阶数字均衡器的传输函数为

$$H(z)=K\frac{a_0z^2+a_1z+a_2}{z^2+b_1z+b_0}$$

式中，

$$a_0=1+Ak_1,\quad a_1=-2\cos(\omega_0),\quad a_2=1-Ak_1$$
$$b_1=-2\cos(\omega_0)/(1+k_1),\quad b_2=(1-k_1)/(1+k_1)$$
$$k_1=\sqrt{(A_c^2-1)/(A^2-A_c^2)}\tan(\Delta\omega/2)$$
$$K=1/(1+k_1)$$

$\Delta\omega$为增益A_c处的带宽，单位为弧度。

(a) 证明当$\omega_0=0$时，传输函数变为一阶低通削减或提升滤波器的传输函数。

(b) 证明当$\omega_0=2\pi f_s/2$时，传输函数变为一阶高通削减或提升滤波器的传输函数。

9.28 生成一个线性调频信号，频率在0.5s内由10Hz变化至20000Hz，抽样频率为44100Hz，将此信号通过两个一阶均衡器。第一个低通提升均衡器增益为4，截止频率为1000Hz，截止频率处的增益应为$1+1/\sqrt{2}$。第二个高通提升均衡器增益为4，截止频率为18000Hz，截止频率处的增益应为$1+1/\sqrt{2}$。画出该信号通过滤波器前后的波形，以及两个均衡滤波器的幅频曲线。

9.29 为验证均衡器处理声音产生的效果，生成一个至少1s的锯齿波，其基频为440Hz，经$f_s=22050\text{Hz}$抽样，带宽限制在小于$f_s/2$的范围内。(可以使用附录D中的fouriersaw函数实现。)使用MATLAB中的sound函数播放原始信号。设计一个峰值均衡器，其中心频率为2200Hz，带宽为3080Hz，峰值增益为4。使用MATLAB中的filter函数将原始信号通过该均衡器，并使用sound函数播放结果。画出锯齿波的频谱以及均衡滤波器的频率响应，利用这两幅图解释原始信号与滤波后信号在听觉上的差别。

9.30 高通提升或削减滤波器的传输函数一般为

$$H(z)=K\frac{z-z_1}{z-p_1}$$

该表示式中存在三个未知量：K、z_1和p_1。对于高通提升滤波器，写出用于求解这些未知量的三个方程，其中最大增益为A，截止频率f_c处的增益为A_c。求解这三个方程并验证表9.3中的值。

第10章
二维数字信号处理简介

许多一维信号的处理方法可以扩展应用于二维信号，如图像数据。生活中许多其他类型的数据本身就是二维信号。因为人眼是人重要的感受器官，为了便于观察，我们通常将其转换为图像信号显示。例如，在夜间天气预报节目中，大气压、温度、湿度和降雨量通常以图像的形式显示。同样，电磁场、重力、地震波等数据也常以二维信号表示。

本章将主要介绍傅里叶变换和FIR滤波器在二维信号中的应用。例如，若将图像通过二维低通滤波器，图像中的边缘部分将被平滑。相反，图像通过二维高通滤波器时，图像轮廓将被锐化。此外，通过傅里叶变换，我们可以确定二维图像的频率成分。

10.1 二维信号的表示

一维信号通常采用沿着时间轴以变化的幅度值表示，二维信号则采用网格上的幅度值表示。每一个网格点都对应一个某种变量(如亮度)的抽样值，网格点称为像素(pixel)。网格的坐标轴通常用空间或时间表示。

10.1.1 像素网格

网格的大小以其宽度和高度上像素点个数的乘积表示，通常称为分辨率。例如，分辨率640×480是指该网格具有480条水平线，每条水平线上有640个像素。数码相机的分辨率就常用像素宽度和高度的乘积来表示。

一个像素代表图像中的一个点。如果图像是黑白图像，一个像素可由1位表示。如果是灰度图像，每个像素需用一个数值表示亮度级，这个数值通常是一个8位数，因为计算机的存储单位为8位，同时人眼很难识别超过256灰度级以外的灰度。对于某些数据类型，256灰度级不够表示——彩色图像中每个像素有红、绿、蓝三个分量，每个分量均表示为256级，这样每个像素需用24位。一个典型的图像网格如图10.1所示，每个像素的每条边都与一个像素相邻，4条边共与4个像素相邻，称为该像素的4邻域，以$N4$表示。如果考虑4个对角上相邻的像素，则称为该像素的8邻域，以$N8$表示。许多基于相邻像素的特性去分析某一像素的图像处理算法是非线性的。本章只考虑基于数字信号处理算法的线性处理。

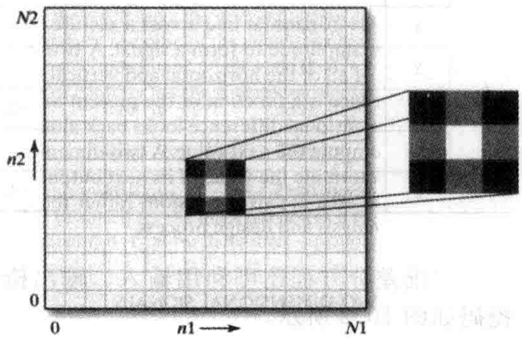


图10.1 典型的$N1\times N2$图像网格。4个浅色阴影的像素是中间白色像素点的$N4$邻域，全部8个阴影像素称为该像素的$N8$邻域

10.1.2 二维差分方程

在数字信号处理中，描述一维因果系统的差分方程可表示为

$$y[n]=b_0x[n]+b_1x[n-1]+\ldots+b_Mx[n-M]-a_1y[n-1]-a_2y[n-2]-\ldots-a_Ny[n-N]$$

与之类似，描述二维因果系统的差分方程可表示为

$$y[k_1,k_2]=\sum_{i_1=0}^{M_1}\sum_{i_2=0}^{M_2}b_{i_1,i_2}x[k_1-i_1,k_2-i_2]-\sum_{\substack{j_1=0\\ j_1,j_2\neq 0,0}}^{N_1}\sum_{j_2=0}^{N_2}a_{j_1,j_2}y[k_1-j_1,k_2-j_2] \tag{10.1}$$

式中，b_{i_1,i_2}，a_{j_1,j_2}表示滤波器系统的系数，x 表示输入信号，y 表示输出信号。滤波器系数和输入、输出信号一样，都是用二维数据描述。二维差分方程的迭代求解相当繁杂，例 10-1介绍了二维差分方程的求解过程。

例 10-1 求解二维差分方程

已知输入是二维的单位脉冲信号，其定义为(0，0)位置的值为 1，其他位置的值为 0。所有初始条件为 0，求解下列差分方程。

$$y[k_1,k_2]=b_{0,0}x[k_1,k_2]-a_{0,1}y[k_1,k_2-1]-a_{1,0}y[k_1-1,k_2]-a_{1,1}y[k_1-1,k_2-1]$$

式中，$b_{0,0}=1$，$a_{0,0}=1$，$a_{0,1}=0.5$，$a_{1,0}=0.5$，$a_{1,1}=0.25$。

解：先令 $k_2=0$，则方程简化为：

$$y[k_1,0]=b_{0,0}x[k_1,0]-a_{0,1}y[k_1,-1]-a_{1,0}y[k_1-1,0]-a_{1,1}y[k_1-1,-1]$$

由于所有初始条件为 0，因而下标为负数的 y 项取值为 0，差分方程进一步化简为

$$y[k_1,0]=b_{0,0}x[k_1,0]-a_{1,0}y[k_1-1,0]$$

根据上述方程，可以很快写出二维信号 y 的第一列

$$y[k_1,0]=[1,-0.5,0.25,-0.125,\cdots$$

同样，可令 $k_2=1，2，\cdots$求解相应的 y。当 k_1、k_2 取值范围为[0，5]时，结果如表 10.1所示。

表 10.1 求解得到的 $y[k_1，k_2]$

		k_2					
		0	1	2	3	4	5
k_1	0	1.0000	−0.5000	0.2500	−0.1250	0.0625	−0.0313
	1	−0.5000	0.2500	−0.1250	0.0625	−0.0313	0.0156
	2	0.2500	−0.1250	0.0625	−0.0313	0.0156	−0.0078
	3	−0.1250	0.0625	−0.0313	0.0156	−0.0078	0.0039
	4	0.0625	−0.0313	−0.0156	−0.0078	0.0039	−0.0020
	5	−0.0313	0.0156	−0.0078	0.0039	−0.0020	0.0010

◀

二维差分方程还可利用输入、输出掩码表示。例 10-1 中差分方程对应的输入、输出掩码如图 10.2 所示。

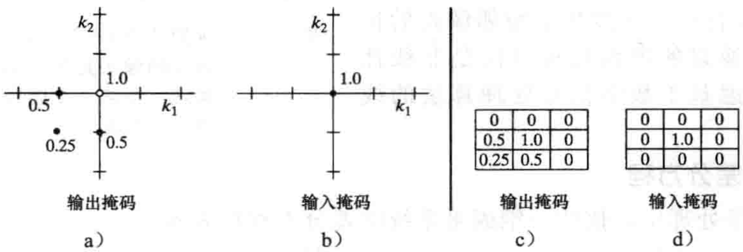


图 10.2 差分方程 $y[k_1，k_2]=x[k_1，k_2]-0.5y[k_1，k_2-1]-0.5y[k_1-1，k_2]-0.25y[k_1-1，k_2-1]$输出和输入掩码的两种描述方式

从输入、输出掩码可以观察到差分方程的求解顺序。由图 10.2 可见，掩码可以沿着相邻较高的行或列连续移动，但由于其左右值还没有计算出来，因而不能在对角线方向移动。此外，还可以构建无法计算的输出掩码，例如，图 10.3 中的输出掩码就不能利用递归法计算得到。

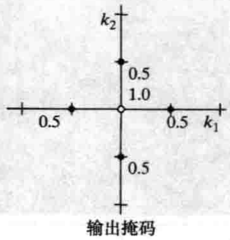


图 10.3　输出掩码不能用递归法求解，因为[0，0]点的输出由[0，1]点和[1，0]点的输出决定，同时，这两点的输出也由[0，0]点的输出决定

10.1.3　二维差分方程的性质

与一维差分方程类似，二维差分方程也有一些重要性质。下面主要介绍其线性、平移不变性、因果性、分离性和稳定性。

线性——如果二维差分方程的输入与某常数相乘后产生的输出等于其原输出的相同常数倍，则该差分方程为线性。在差分方程中，输入的幂运算，自身平移后的相乘或进行布尔运算通常会产生非线性差分方程。

平移不变性——该性质与线性特性类似，如果输入平移后所得输出等于原输出的相同平移，则该差分方程具有平移不变性。若有

$$y[k_1,k_2]=G(x[k_1,k_2])$$

对于平移不变性的系统

$$y[k_1-n_1,k_2-n_2]=G(x[k_1-n_1,k_2-n_2])$$

n_1，n_2 为任意整数。

因果性——如果 k_1 或 k_2 小于零时，二维差分方程所描述系统的脉冲响应为零，则该二维差分方程具有因果性。其必要条件为：$h[k_1,\ k_2]=0$，其中 $k_1<0$ 或 $k_2<0$。

二维系统的因果性与一维系统的因果性稍有不同。在一维系统中，x 轴通常是指时间轴，具有因果性的一维系统随输入信号的时间顺序而变化；在二维系统中，表示图像的两个维度都是空间变量，因而是否具有因果性对于系统的作用并不如一维系统重要。

分离性——一维差分方程没有此类特性。如果一个二维差分方程的响应可以表示为两个一维响应的乘积，则称二维差分方程具有分离性。构造分离性差分方程并不难，可以将两个一维差分方程的响应相乘得到一个二维差分方程的响应。例如，假设两个一维差分方程的响应分别如下

$$f[k_1]=\begin{cases}0 & k_1<0\\ \mathrm{e}^{-k_1/10} & k_1\geqslant 0\end{cases}$$

$$g[k_2]=\begin{cases}0 & k_2<0\\ 1-\mathrm{e}^{k_2/10} & k_2\geqslant 0\end{cases}$$

从而得到二维差分方程的响应

$$y[k_1,k_2]=g[k_2]\times f[k_1]$$

上述构造 $y[k_1, k_2]$ 的过程如图10.4所示。其中，颜色越浅，对应的灰度值越大。

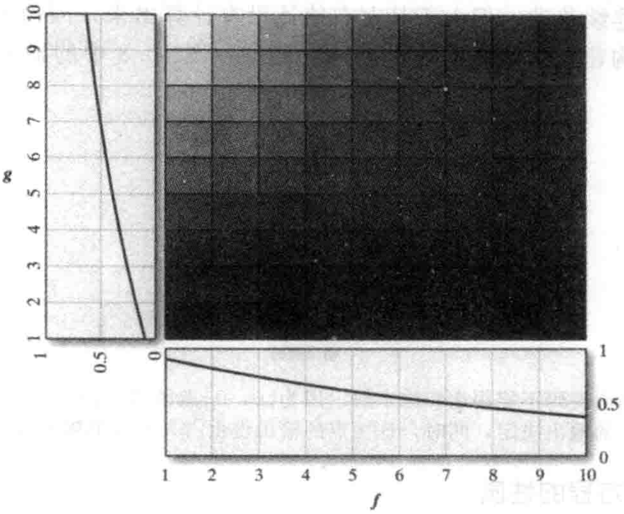


图10.4 两个一维函数相乘构建一个二维函数的过程

二维方程的分离性可以应用于其他运算，如二维卷积可分解成一维运算。

稳定性——如果二维系统脉冲响应的绝对值之和为有限值，则该二维系统具有稳定性，即

$$\sum_{k_1=-\infty}^{\infty}\sum_{k_2=-\infty}^{\infty}|h[k_1,k_2]|<\infty$$

上述表示式虽然简单，但实际应用很困难。二维系统不能像一维系统那样通过判断极点是否位于单位圆内来判断系统的稳定性。我们将在介绍二维 z 变换和二维传输函数后，再讨论二维系统的稳定性。

10.2 二维变换

10.2.1 二维 z 变换

二维 z 变换定义为

$$Z\{x[k_1,k_2]\}=X(z_1,z_2)=\sum_{k_1=-\infty}^{\infty}\sum_{k_2=-\infty}^{\infty}x[k_1,k_2]z_1^{-k_1}z_2^{-k_2} \tag{10.2}$$

相应的 z 反变换为

$$x[k_1,k_2]=\frac{1}{(2\pi j)^2}\iint_{C_1C_2}X(z_1,z_2)z_1^{k_1-1}z_2^{k_2-1}dz_1dz_2$$

式中，C_1，C_2 是闭合积分曲线。与一维 z 反变换一样，二维 z 反变换的计算较复杂且很少应用。

例10-2 二维信号的 z 变换

求下列二维信号的 z 变换

$$x[0,0]=1,\quad x[0,1]=x[1,0]=0.5,\quad x[1,1]=0.25$$

解： 由式(10.2)可得

$$X(z_1,z_2)=1+0.5z_1^{-1}+0.5z_2^{-1}+0.25z_1^{-1}z_2^{-1}$$ ◀

卷积。若线性二维系统的脉冲响应为 $h[k_1,k_2]$，输入信号为 $x[k_1,k_2]$，则输出 $y[k_1,k_2]$ 等于 $h[k_1,k_2]$ 与 $x[k_1,k_2]$ 的卷积，即

$$y[k_1,k_2] = x[k_1,k_2] * h[k_1,k_2] = h[k_1,k_2] * x[k_1,k_2]$$

式中，$*$ 是卷积符号。一般情况下，二维卷积运算表示为

$$y[k_1,k_2] = \sum_{n_1=-\infty}^{\infty}\sum_{n_2=-\infty}^{\infty} h[n_1,n_2]x[k_1-n_1,k_2-n_2] \tag{10.3}$$

利用式(10.2)，将上式两边同时进行 z 变换，得到

$$Y(z_1,z_2) = \sum_{k_1=-\infty}^{\infty}\sum_{k_2=-\infty}^{\infty}\sum_{n_1=-\infty}^{\infty}\sum_{n_2=-\infty}^{\infty} (h[n_1,n_2]z_1^{-n_1}z_2^{-n_2})(x[k_1-n_1,k_2-n_2]z_1^{-(k_1-n_1)}z_2^{-(k_2-n_2)})$$

交换求和次序得

$$Y(z_1,z_2) = \sum_{n_1=-\infty}^{\infty}\sum_{n_2=-\infty}^{\infty}(h[n_1,n_2]z_1^{-n_1}z_2^{-n_2})\sum_{k_1=-\infty}^{\infty}\sum_{k_2=-\infty}^{\infty} (x[k_1-n_1,k_2-n_2]z_1^{-(k_1-n_1)}z_2^{-(k_2-n_2)})$$

上式可化简为

$$Y(z_1,z_2) = H(z_1,z_2) \times X(z_1,z_2) \tag{10.4}$$

式中，$H(z_1,z_2)$为二维系统的传输函数，也是系统脉冲响应 $h[k_1, k_2]$的 z 变换。根据二维差分方程式(10.1)，传输函数可写为

$$H(z_1,z_2) = \frac{\sum\limits_{k_1=0}^{M_1}\sum\limits_{k_2=0}^{M_2} b(k_1,k_2)z_1^{-k_1}z_2^{-k_2}}{\sum\limits_{k_1=0}^{N_1}\sum\limits_{\substack{k_2=0 \\ k_1,k_2\neq 0,0}}^{N_2} a(k_1,k_2)z_1^{-k_1}z_2^{-k_2}} \tag{10.5}$$

例 10-3 二维差分方程的 z 变换

求例 10-1 中二维差分方程

$y[k_1, k_2]=b_{0,0}x[k_1, k_2]-a_{0,1}y[k_1, k_2-1]-a_{1,0}y[k_1-1, k_2]-a_{1,1}y[k_1-1, k_2-1]$的传输函数 $H(z_1,z_2)$。

解：对差分方程两边同时进行二维 z 变换

$$Y(z_1,z_2) = b_{0,0}X(z_1,z_2) - a_{1,0}z_1^{-1}Y(z_1,z_2) - a_{0,1}z_2^{-1}Y(z_1,z_2) - a_{1,1}z_1^{-1}z_2^{-1}Y(z_1,z_2)$$

化简得到

$$H(z_1,z_2) = \frac{Y(z_1,z_2)}{X(z_1,z_2)} = \frac{b_{0,0}}{1+a_{1,0}z_1^{-1}+a_{0,1}z_2^{-1}+a_{1,1}z_1^{-1}z_2^{-1}}$$ ◀

注意，在一维情况下，IIR 滤波器系统函数的分子、分母多项式可表示为零极点形式，若极点全部位于 z 平面单位圆内，则系统是稳定系统。在二维情况下，系统函数难以表示为零极点形式，从 z 域判断系统的稳定性更加困难。

在一维 z 变换中，z 是一个复数变量，可将其看成由实部和虚部构成的二维数据。在二维情况下，有两个独立的 z 变量 z_1，z_2，两者均由实部和虚部构成，因此二维 z 变换有 4 个维度，或者说有 4 个自由度。一维情况下的单位圆，在二维中变为单位圆面，在三维中变为单位球，在四维情况下将变得很抽象。对于零极点，一维信号处理中其为点，而在二维信号处理中变成了面。

下面介绍两个关于二维 z 变换稳定性的定理，其证明不在此进行阐述[1]。

定理一：假设二维系统为 $H(z_1, z_2)=1/(A(z_1, z_2))$，如果

$$A(z_1,z_2) \neq 0, \quad |z_1| \geqslant 1 \text{ 且 } |z_2| \geqslant 1$$

则该二维系统稳定。

注意，$H(z_1,z_2)$的分子是1而不是二维多项式$B(z_1,z_2)$，这是因为在二维情况下，零点所在平面很有可能与极点平面相交，从而零点和极点相互抵消。对于二维系统$H(z_1,z_2)=B(z_1,z_2)/A(z_1,z_2)$，这个定理是系统稳定的充分而非必要条件。

定理二：假设二维系统为$H(z_1,z_2)=1/(A(z_1,z_2))$，如果

$$A(z_1,z_2)\neq 0,|z_1|=1\text{ 且 }|z_2|\geqslant 1;A(z_1,z_2)\neq 0,|z_1|\geqslant 1\text{ 且 }|z_2|=1$$

则该二维系统稳定。

对于IIR滤波器的二维传输函数，上述两个定理均无法提供一个简便的方法来判断系统的稳定性。

频率响应。在一维情况下，可以用$e^{j\omega T}$代替z得到系统的频率响应。在二维系统中有两个抽样频率和两个ω值，频率响应可写为

$$H(e^{j\omega_1 T_1},e^{j\omega_2 T_2})=\frac{\sum_{k_1=0}^{M_1}\sum_{k_2=0}^{M_2}b(k_1,k_2)e^{-j\omega_1 T_1 k_1}e^{-j\omega_2 T_2 k_2}}{\sum_{k_1=0}^{N_1}\sum_{\substack{k_2=0\\ k_1,k_2\neq 0,0}}^{N_2}a(k_1,k_2)e^{-j\omega_1 T_1 k_1}e^{-j\omega_2 T_2 k_2}}$$

例 10-4　二维FIR滤波器的频率响应

利用MATLAB绘制二维FIR滤波器的频率响应图，滤波器的传输函数为

$$H(z_1,z_2)=1+1.4(z_1^{-1}+z_2^{-1})+0.7(z_1^{-2}+z_2^{-2})+1.96z_1^{-1}z_2^{-1}+0.98z_1^{-1}z_2^{-2}+0.98z_1^{-2}z_2^{-1}+0.49z_1^{-2}z_2^{-2}$$

解：将$H(z_1,z_2)$写成二维数组形式，如下表所示。

n_2 \ n_1	0	1	2	3	4
4	—	—	—	—	—
3	0	0	0	0	—
2	0.7	0.98	0.49	0	—
1	1.4	1.96	0.98	0	—
0	1	1.4	0.7	0	—

该系数矩阵h为二维FIR滤波器的脉冲响应。针对该问题，我们可以构建行向量$h1=\{1, 1.4, 0.7, 0, 0, \cdots\}$乘以其转置后的列向量$h2$得到系数矩阵$h$。上述过程的MATLAB代码如下：

```
N = 65;
h1 = zeros(1, N);
h1(1) = 1;
h1(2) = 1.4;
h1(3) = 0.7;
h2 = h1;
h = h1'*h2;
```

该滤波器为二维FIR滤波器，可以调用freqz2函数绘制频率响应图。

```
figure(1);clf;
freqz2(h, N, N);
```

程序运行结果如图10.5所示。

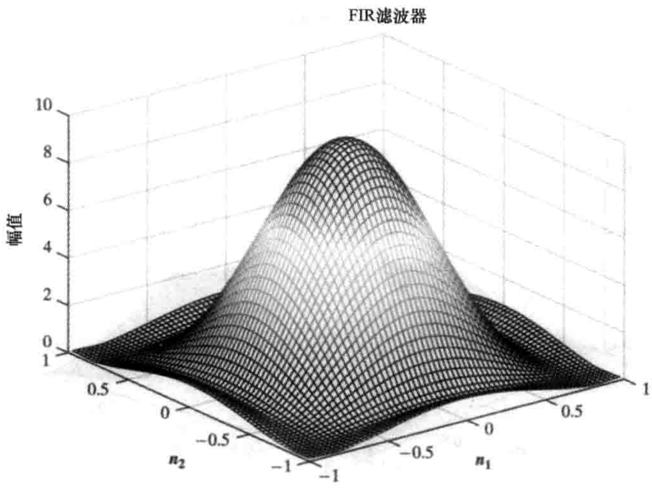


图 10.5　调用 freqz2 函数绘制的二维 FIR 滤波器的频率响应图，其中 MATLAB 的频率响应中心为(0，0)点

例 10-5　二维 IIR 滤波器的频率响应

利用 MATLAB 绘制二维 IIR 滤波器的频率响应图，该滤波器的传输函数为

$$H(z_1,z_2)=\frac{1+0.5z_1^{-1}+0.5z_2^{-1}-0.25z_1^{-1}z_2^{-1}}{1-0.5z_1^{-1}-0.5z_2^{-1}+0.25z_1^{-1}z_2^{-1}}$$

解： MATLAB 提供了 freqz2 函数可用来绘制二维 FIR 滤波器的频率响应图，由于这是一个 IIR 滤波器，因而可以将传输函数的分子、分母分离并视为独立的 FIR 滤波器。IIR 滤波器的幅频响应等于分子的幅值与分母的幅值之比。

下面的 MATLAB 程序定义分子函数和分母函数。

```
N1 = 64;N2 = 64;
h1 = zeros(N1, N2); %Denominator
h1(1, 1) = 1;
h1(1, 2) = -.5;
h1(2, 1) = -.5;
h1(2, 2) = .25;
h2 = zeros(N1, N2); %Numerator
h2(1, 1) = 1;
h2(1, 2) = .5;
h2(2, 1) = .5;
h2(2, 2) = -.25;
```

调用 freqz2 函数分别得到分子、分母多项式的幅值。

```
figure(1);clf;
[H1 f1 f2] = freqz2(h1, N1, N2);
[H2 f1 f2] = freqz2(h2, N1, N2);
mesh(f1, f2, abs(H2)./abs(H1));
```

程序运行后的结果如图 10.6 所示。

10.2.2　二维离散傅里叶变换

二维 DFT 可看作一维 DFT 的扩展，它将图像表示成由多个不同频率的正弦信号构成。二维 DFT 及其反变换见式(10.6)和式(10.7)。

2D DFT
$$X(k_1,k_2)=\sum_{n_1=0}^{N_1-1}\sum_{n_2=0}^{N_2-1}x[n_1,n_2]\mathrm{e}^{-\mathrm{j}(2\pi/N_1)n_1k_1}\,\mathrm{e}^{-\mathrm{j}(2\pi/N_2)n_2k_2}\tag{10.6}$$

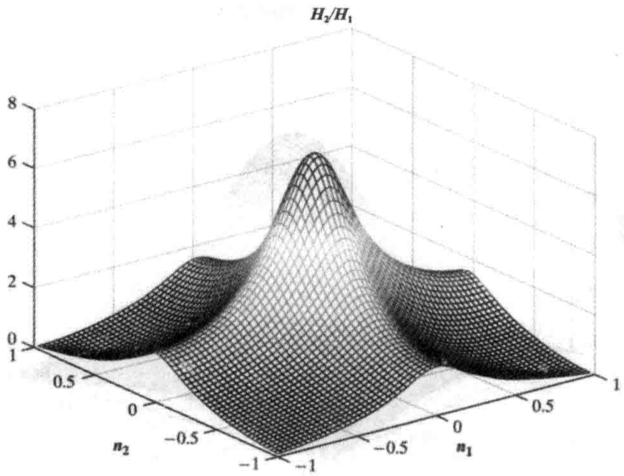


图 10.6 二维 IIR 滤波器的幅频响应图

式中，$0\leqslant k_1\leqslant N_1-1$，$0\leqslant k_2\leqslant N_2-1$

2D IDFT $$x[n_1,n_2]=\frac{1}{N_1N_2}\sum_{k_1=0}^{N_1-1}\sum_{k_2=0}^{N_2-1}X(k_1,k_2)\mathrm{e}^{\mathrm{j}(2\pi/N_1)n_1k_1}\mathrm{e}^{\mathrm{j}(2\pi/N_2)n_2k_2} \tag{10.7}$$

式中，$0\leqslant n_1\leqslant N_1-1$，$0\leqslant n_2\leqslant N_2-1$

由以上两式可知，$N_1\times N_2$ 点的二维序列 $x[n_1,n_2]$对应频域中的另一个 $N_1\times N_2$ 点的二维序列，两个序列均定义在第一象限。因此，在某些情况下，为了应用二维离散傅里叶变换，需将有限序列平移至第一象限。在频域内，零频率点位于原点处，其他更高的频率采用以原点为中心的同心圆表示。与一维情况类似，二维离散傅里叶变换以 N_1、N_2 为周期，周期性可用下列方程表示

$$\mathrm{Re}[X(k_1,k_2)]=\mathrm{Re}[X(N_1-k_1,N_2-k_2)]$$
$$\mathrm{Im}[X(k_1,k_2)]=-\mathrm{Im}[X(N_1-k_1,N_2-k_2)]$$

计算 2D DFT 时，取图像的左下角为原点，但在显示 2D DFT 时，原点往往移到图像的中心。

式(10.6)和式(10.7)的计算复杂度很高，但正如一维的情况，这两个公式都存在相当多的计算冗余。在式(10.6)中，可以从第二个求和项中将不包含 k_2 的项提取出来，式(10.6)可写为

$$X(k_1,k_2)=\sum_{n_1=0}^{N_1-1}\mathrm{e}^{-\mathrm{j}(2\pi/N_1)n_1k_1}\sum_{n_2=0}^{N_2-1}x[n_1,n_2]\mathrm{e}^{-\mathrm{j}(2\pi/N_2)n_2k_2}$$

上式化简为

$$X(k_1,k_2)=\sum_{n_1=0}^{N_1-1}Y(n_1,k_2)\mathrm{e}^{-\mathrm{j}(2\pi/N_1)n_1k_1} \tag{10.8}$$

式中，$Y(n_1,\ k_2)$是 N_2 点的 1D DFT。

$$Y(n_1,k_2)=\sum_{n_2=0}^{N_2-1}x[n_1,n_2]\mathrm{e}^{-\mathrm{j}(2\pi/N_2)n_2k_2} \tag{10.9}$$

如果指定 n_2 为行变量，n_1 为列变量，则 $Y(n_1,k_2)$表示第 n_1 列的 DFT。

根据式(10.8)和式(10.9)，可以通过以下步骤计算 $X(k_1,k_2)$：

1. 计算 $x[n_1,n_2]$中每一列的 DFT $Y(n_1,k_2)$。
2. 计算 $Y(n_1,k_2)$中每一行的 DFT。

同样，可以用类似步骤计算二维离散傅里叶反变换。

例 10-6 指数衰减的二维正弦信号的 DFT

利用 MATLAB 计算如下所示的指数衰减的二维正弦信号的 DFT

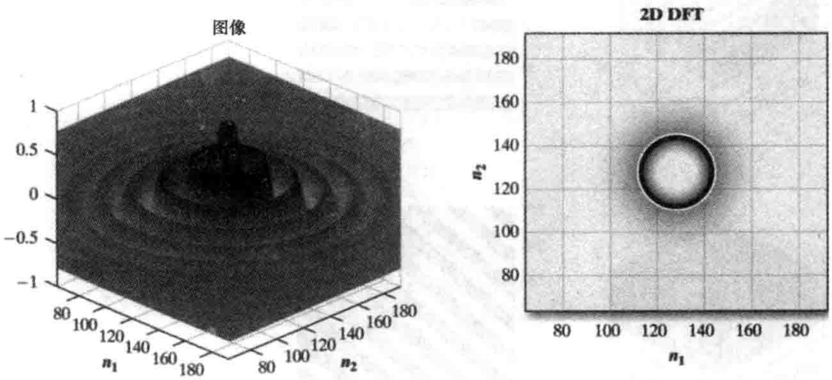
$$y[n_1, n_2] = \mathrm{e}^{-r/\tau}\sin(2\pi fr)$$

式中，r 是到原点的距离，$\tau=0.15$，$f=8\text{Hz}$。令 $N=256$，平移图像使其原点位于第一象限的中心点，绘制图像并计算其 DFT。

解：第一象限中，n_1、n_2 的范围都为[0，255]。图像中心点为$(n_1, n_2)=(128，128)$，平移后的 r 可表示为

$$r = \sqrt{(n_1 - N/2)^2 + (n_2 - N/2)^2}$$

下面的 m 文档结合了 MATLAB 中的图像处理工具箱进行编写。程序运行后得到的图像和 2D DFT 的结果如图 10.7 所示。注意，DFT 结果以俯视图显示，而程序运行得到的图像是以三维 30°仰角显示。



a）$y[n_1, n_2]=\mathrm{e}^{-r/\tau}\sin(2\pi fr)$ 的图像　　b）2D DFT结果的俯视图，函数plot3d和invertcolormap见附录D

图 10.7 例 10-6 图

```
%fftDecaySine.m
N = 256;
imap = invertcolormap(colormap(gray));
f = 16;
tau = .1;
xc = N/2;yc = N/2;
r = ones(N, N);
for n1 = 1:N
    for n2 = 1:N
       r(n1, n2) = sqrt((xc-n1)^2 + (yc-n2)^2);
    end
end
W = 2*pi*f*r/N;
%create image
im = exp(-(r/N)/tau).*sin(W);
ymag = abs(fft2(im)); %2d fft
ymag = ymag/abs(max(max(ymag))); %normalize
figure(1);clf;
N = size(im, 1);
b = N/4;
axis3d = [N/2-b N/2+b N/2-b N/2+b -1 1];
plot3d(im, N, 1, imap, [45 30], axis3d);
```

```
title('Image')
figure(2);clf;
plot3d(fftshift(ymag), N, 2, imap, [0 90],axis3d)
title('2D DFT');
```

◀

图 10.8 给出了一些常用信号的 2D DFT 结果。每个信号都在第一象限内显示，其 DFT 为平移后的结果，以使原点位于图像的中心。可以看到，水平方波和对角方波的基波以及奇次谐波显示为与该方波方向相同的直线。对于圆形方波，进行 DFT 后，其谐波频率以基波为中心展开成同心圆。对于正弦方波，其 DFT 后二维频率空间为一个圆。频率越高，圆的直径越大。原点表示其直流分量，在本例中直流分量为零。

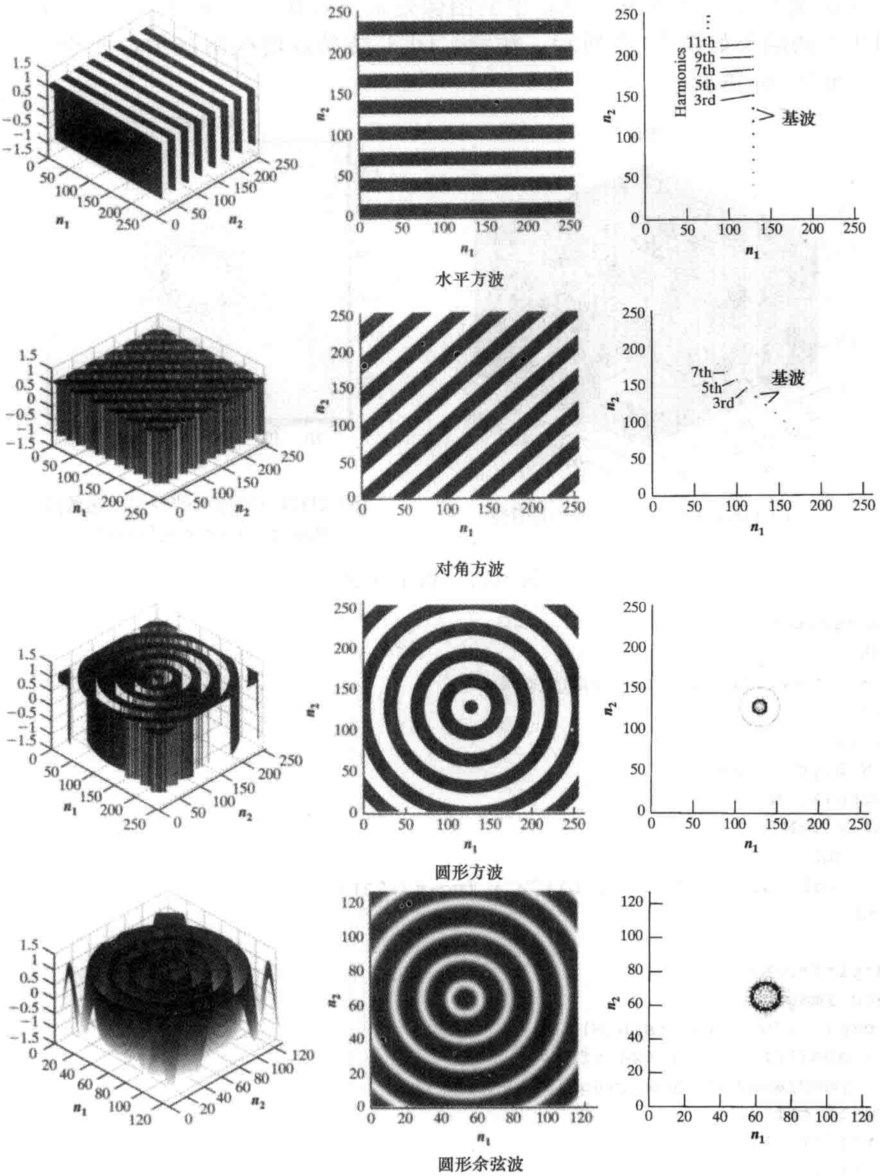


图 10.8　不同波形的二维 DFT，变换结果进行了平移使得零频率分量位于图像的中心

10.2.3　2D DFT 的性质

2D DFT 的性质与一维离散傅里叶变换类似，见表 10.2。x 和 y 用来表示一个二维信号，自变量 n_1 的取值定义在$[0, N_1-1]$区间，自变量 n_2 的取值定义在$[0, N_2-1]$区间，其对应的 2D DFT 分别为 X 和 Y，可写为 $x \Leftrightarrow X$，$y \Leftrightarrow Y$。

表 10.2　2D DFT 的性质

线性	$ax+by \Leftrightarrow aX+bY$，其中 a、b 为常数
循环卷积	x、y 的循环卷积对应其各自 DFT 的乘积，即 $x \otimes y \Leftrightarrow X \cdot Y$
时域乘积	x、y 的乘积对应 X 和 Y 循环卷积的常数倍，即 $x \cdot y \Leftrightarrow \frac{1}{N_1 N_2} X \otimes Y$
可分性	如果 x 可写成两个序列的乘积，则 x 的 DFT 也是两个序列各自 DFT 的乘积 即，$x = x_1[n_1] \cdot x_2[n_2] \Leftrightarrow X_1(k_1) \cdot X_2(k_2)$
循环移位	X 具有周期性，所以可以循环形式平移，即 $x[i, j] \Leftrightarrow X(k_1, k_2) e^{-j2\pi(k_1 m_1/N_1 + k_2 m_2/N_2)}$ 式中，$i = (n_1 - m_1) \bmod N_1$， $j = (n_2 - m_2) \bmod N_2$
帕斯瓦尔定理	$\sum_{n_1=0}^{N_1-1} \sum_{n_2=0}^{N_2-1} x \cdot y^* = \frac{1}{N_1 N_2} \sum_{k_1=0}^{N_1-1} \sum_{k_2=0}^{N_2-1} X \cdot Y^*$ 若 $x=y$，则等式左边表示信号的能量 $\sum_{n_1=0}^{N_1-1} \sum_{n_2=0}^{N_2-1} \lvert x \rvert^2 = \frac{1}{N_1 N_2} \sum_{k_1=0}^{N_1-1} \sum_{k_2=0}^{N_2-1} \lvert X \rvert^2$
对称性	(a)$x^*[n_1, n_2] \Leftrightarrow X^*(-k_1, -k_2)$ (b)$X(k_1, k_2) = X^*(i, j)$，式中，$i = -k_1 \bmod N_1$， $j = -k_2 \bmod N_2$ 从而得到 实部 $\mathrm{Re}(X)$是偶函数，虚部 $\mathrm{Im}(X)$是奇函数。 模 $\lvert X \rvert$ 是偶函数，相角 θ_X 是奇函数

10.2.4　2D DFT 与卷积

二维卷积方程如式(10.3)所示，两幅图像卷积的难点在于计算的复杂性。利用 DFT 的卷积性质可以极大地降低算法的复杂度，即首先采用 FFT 算法实现 DFT，然后计算 DFT 的乘积从而获得两幅图像的卷积。卷积特性表明由于 DFT 的周期性，我们得到的结果为循环卷积。若要计算线性卷积，需要对信号进行补零。例如，假如信号 x 在 n_1 方向的范围为 $0 \sim L_1-1$，信号 y 在 n_1 方向的范围为 $0 \sim M_1-1$，卷积后的信号在 n_1 方向的长度为 $N_1 = L_1 + M_1 - 1$，为了计算线性卷积，需要先将两个信号补零至卷积后信号的长度。

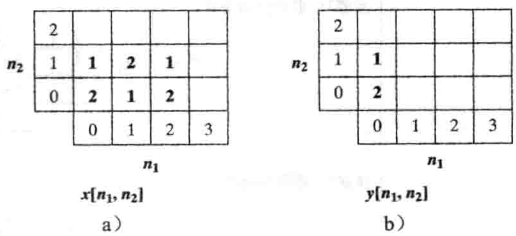


图 10.9　信号 $x[n_1, n_2]$和 $y[n_1, n_2]$

例 10-7　利用 DFT 计算线性卷积

利用 DFT 的卷积性质计算图 10.9 中两个信号的线性卷积。

解：由式(10.3)，得到

$$z[k_1, k_2] = \sum_{n_1=-\infty}^{\infty} \sum_{n_2=-\infty}^{\infty} x[n_1, n_2] y[k_1 - n_1, k_2 - n_2]$$

首先计算二维信号的图形卷积，将信号 y 的 n_1、n_2 坐标进行翻转，如图 10.10 所示。

然后，对应每一对(k_1,k_2)，平移信号 y 的(0,0)点，然后与 x 相乘，对乘积求和得到 $z[k_1,k_2]$。$[k_1,k_2]=[1,1]$的计算过程如图 10.11 所示。

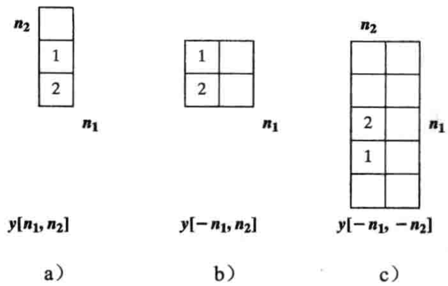


图 10.10　将信号 y 的两个变量翻转，为图形卷积做准备

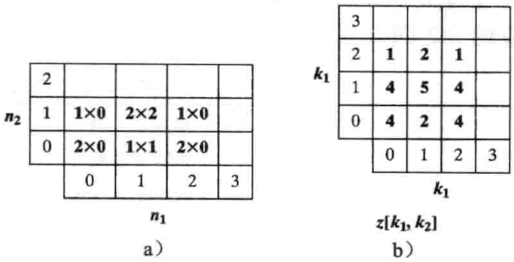


图 10.11　计算 $z[1,1]$的值，先将翻转后的 y 信号平移，使其[0，0]点移至[1，1]点，再与 x 的对应项相乘，最后求和得到 $z[1,1]=5$。继续平移翻转后的 y 信号，将乘积结果求和得到最终的卷积结果如 b 所示

若要通过 DFT 得到上述相同结果，首先对每个信号适当补零。本例中，x 是 2×3 的信号，y 是 2×1 的信号，则 DFT 后大小为 3×3。接下来分别对 x、y 进行 2D DFT 得到 $X(k_1,k_2)$和 $Y(k_1,k_2)$。最后，将 X 和 Y 的乘积进行 2D IDFT。图 10.12 所示为每一步的结果。

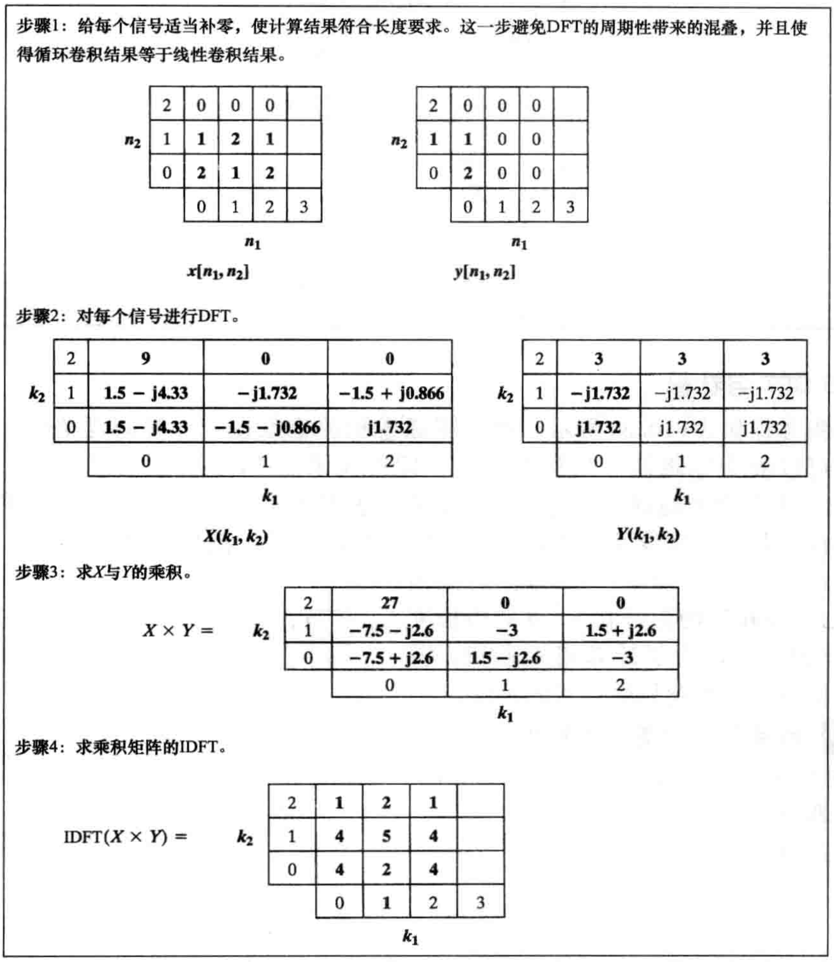


图 10.12　利用 DFT 卷积性质计算线性卷积的过程

10.2.5　2D DFT 与光学

傅里叶光学是二维傅里叶变换理论在光的传播方面的应用。傅里叶变换可用于描述多种类型的衍射波，目前在图像处理、透镜性能和参数、光信号处理和光计算中已有实际应用。

图 10.13 所示光学系统中，入射光为平面波光线，穿过物体并通过第一个透镜后聚焦，在第二个透镜处成像。由图 10.13 可以看出，聚焦平面为物体的二维傅里叶变换平面[4]。

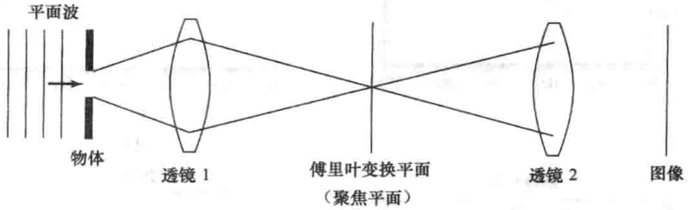


图 10.13　聚焦平面对目标进行二维傅里叶变换，第二个透镜进行傅里叶反变换生成图像

该过程与直觉相一致，沿 z 轴传播的平面波表示为复指数形式 $\psi(z)=Ae^{\pm j2\pi z/\lambda}$，$\lambda$ 表示波长，A 为幅值常数。聚焦面上任意点接收来自图像的所有光线，并且可以表示为复指数函数的加权和，其与傅里叶变换形式相同。图 10.13 中的第二个透镜对聚焦平面进行傅里叶反变换以对图像进行恢复。

如果将透明薄片放置在傅里叶变换平面，可以在希望阻止的频率处设置为不透明来滤除 2D DFT 后的某些频率。例如，将傅里叶变换平面中的中心点变黑可以削弱低频分量，傅里叶变换平面中的黑色环可以削弱高频分量。图 10.14 所示为 3 幅图像通过与图 10.13 相似的透镜仿真系统后计算获得的图像，这些计算得到的图像和相似条件下经过透镜所产生的图像非常接近。

例 10-8　二维图像的手动滤波

下面的 MATLAB 程序运行后结果如图 10.15 所示。水平和垂直条纹宽度的变化使得图像的 2D DFT 具有沿着 n_1、n_2 轴的不同频率分量。设计如何手动修改图像的 DFT 从而实现截止频率为 $\pi/8$ 的低通滤波。

```
N = 256;
im = zeros(N, N);
stp = 16;
for i = 1:N/stp-1
  strt = i*stp - stp + 1;
  im(:,strt:strt+i*stp/(N/stp)) = 1;
  im(strt:strt+i*stp/(N/stp), :) = 1;
end
```

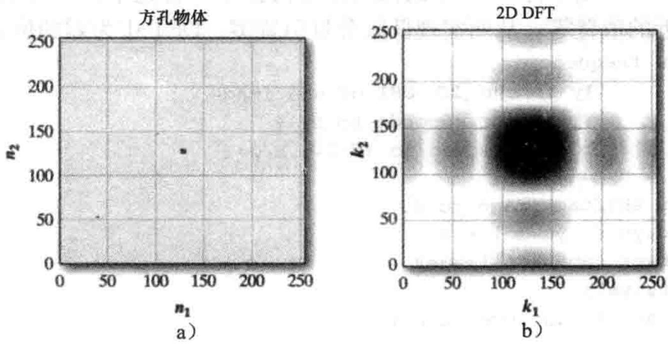


图 10.14　左边为图像，右边为图像的傅里叶变换

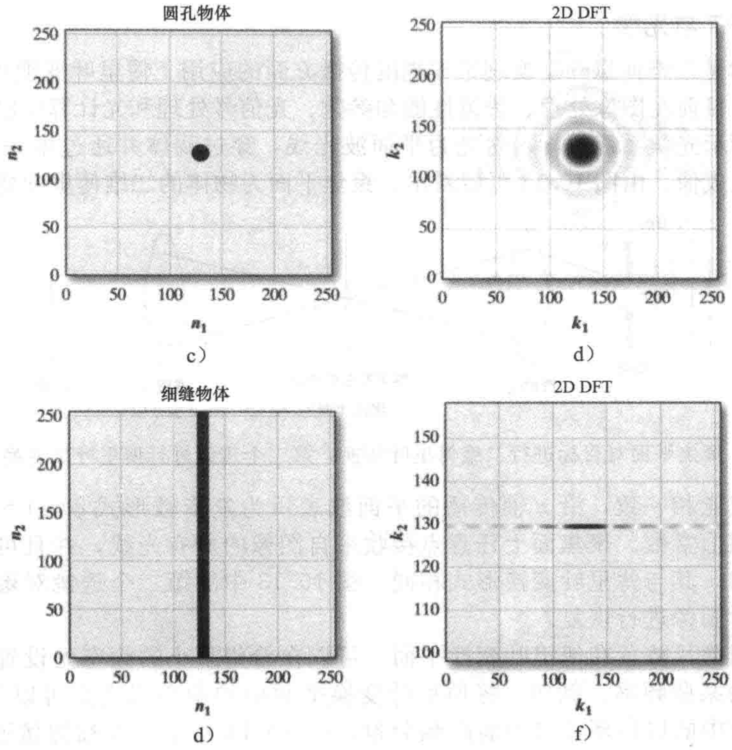


图 10.14 （续）

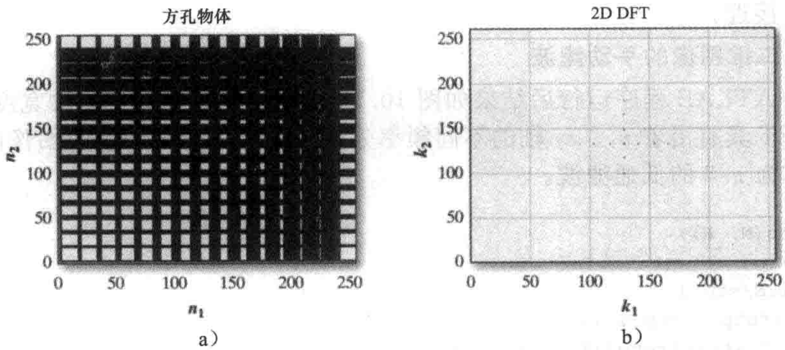


图 10.15 图像及其基于二维离散傅里叶变换的滤波

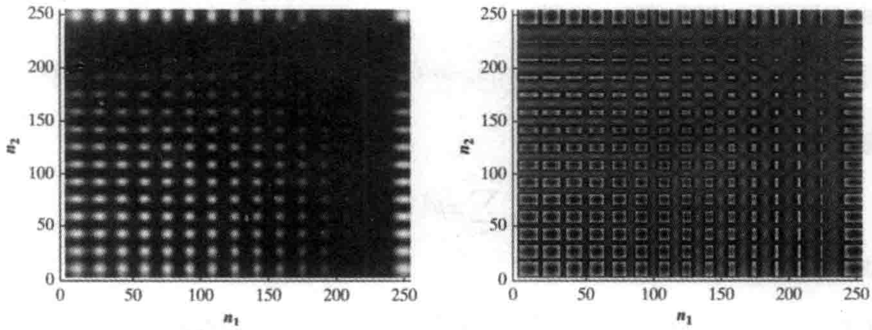
解： 下面的 MATLAB 程序将 DFT 中的低频分量值置零。首先平移图像的 DFT 结果使其居中，然后将中心附近的值置零，从而实现低频分量值置零。DFT 中为零的值其共轭也为零。

```
% Kill off low frequencies
y = fft2(im);       %y is the 2D DFT of the image
fc = N/16;          %pi/8 corresponds to N/16
ys = fftshift(y);   %Center moved to (N/2+1,N/2+1)
xc = N/2+1;yc = N/2+1;
%Set center of shifted image to 0.
ys(xc-fc:xc+fc,yc-fc:yc+fc) = 0;
%Shift image back "unshift image"
iys = ifftshift(ys);
imT = ifft2(iys); %Find inverse DFT
%Plot result
plot3d(imT, N, 3, imap);
```

如果希望低频分量通过，先构建一个全零的图像，然后使用 DFT 中的数值填充低频部分：

```
% Kill off high frequencies
ys = fftshift(y); %Center moved to (N/2+1,N/2+1)
xc = N/2+1;yc = N/2+1;
yhigh = zeros(N, N); %Create an image of all zeros
%Fill in the low frequencies
yhigh(xc-fc:xc+fc,yc-fc:yc+fc) = ys(xc-fc:xc+fc,yc-fc:yc+fc);
iys = ifftshift(yhigh);
imT = ifft2(iys);
plot3d(imT, N, 4, imap);
```

plot3d 函数的使用说明见附录 D，本例运用 MATLAB 图像处理工具箱，手动滤波结果如图 10.16 所示。



a）低通滤波结果　　b）高通滤波结果

图 10.16　滤波结果

10.2.6　二维离散余弦变换

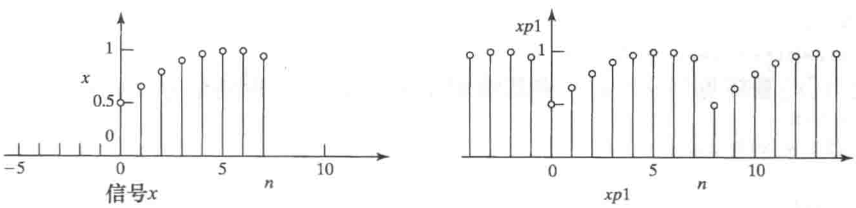
Joseph Fourier 证明了任意周期函数可表示为无限个离散频率的正弦和余弦函数之和。傅里叶变换来源于傅里叶级数，并用于分析非周期信号。在傅里叶变换中，时间和频率都是连续变量。DFT 来源于傅里叶变换，其时间和频率都是离散变量，可使用计算机进行运算。DFT 使用正弦和余弦函数作为基函数，从而可用欧拉公式将其转换成复指数形式。由于 DFT 同时在时域和频域上抽样，因此原始信号和变换后的信号都是周期的。

假设信号为

$$x[n]=\sin\left(\frac{\pi n}{2N}+\frac{\pi}{6}\right),\quad 式中\ N=8,n=0,1,2,\cdots,7$$

信号如图 10.17a 所示。

对 $x[n]$进行 DFT 得到 $X(k)$，$X(k)$为周期函数。对 $X(k)$进行 IDFT，得到周期化的 $x[n]$，用 $xp1[n]$表示，如图 10.17b 所示。



a）$x[n]$是8点实信号　　b）$xp1[n]$是周期信号

图 10.17　对 $x[n]$进行 DFT 得到 $X(k)$，反变换后得到 $x[n]$的周期延拓 $xp1[n]$

但是 DFT 产生的周期性导致周期端点上产生突变，因而在信号的高频处引入更多的

能量。图10.18介绍了另一种周期化的方法，即将信号 $x[n]$ 翻转后附加在原信号 $x[n]$ 之后，然后再周期化，得到信号 $xp[n]=xp[2N-n-1]$，这样端点变得更平滑，从而减少了高频信号的能量。

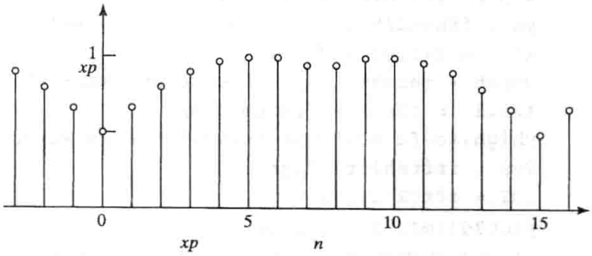


图10.18 将信号 $x[n]$ 翻转后附加在原信号 $x[n]$ 之后，然后周期化，注意 $xp[n]=xp[2N-n-1]$

利用信号的对称性，[0，2N−1]上的 $xp[n]$ 的DFT可写为

$$XP(k)=\sum_{n=0}^{N-1}xp[n]\mathrm{e}^{-\mathrm{j}2\pi k/(2N)}+\sum_{n=0}^{N-1}xp[n]\mathrm{e}^{-\mathrm{j}2\pi k/(2N-n-1)/(2N)}$$

在上式中提出 $\mathrm{e}^{-\mathrm{j}(2\pi k/2)/(2N)}$，得到

$$XP(k)=\mathrm{e}^{-\mathrm{j}(2\pi k/2)/(2N)}\left[\sum_{n=0}^{N-1}xp[n]\mathrm{e}^{-\mathrm{j}2\pi(k/2)/(2N)}+\sum_{n=0}^{N-1}xp[n]\mathrm{e}^{-\mathrm{j}2\pi k(n+1/2)/(2N)}\right]$$

利用欧拉公式将上式化简为

$$XP(k)=2\mathrm{e}^{-\mathrm{j}(2\pi k/2)/(2N)}\left[\sum_{n=0}^{N-1}xp[n]\cos(\pi k(2n+1)/(2N))\right]$$

单边离散余弦变换(DCT)定义为

1D DCT
$$C(k)=p(k)\sum_{n=0}^{N-1}x[n]\cos(\pi k(2n+1)/(2N)) \tag{10.10}$$

式中，

$$p(k)=\begin{cases}\sqrt{1/N}, & k=0\\ \sqrt{2/N}, & k=1,2,\cdots,N-1\end{cases}$$

DCT不是DFT的实部，但可通过以下步骤从DFT得到DCT：

1. 将 $x[n]$ 进行周期延拓得到 $x[n]=x[2N-n-1]$。
2. 对周期信号进行离散傅里叶变换得到 $X(K)$。
3. 将 $X(K)$ 与 $p(k)\mathrm{e}^{\mathrm{j}(2\pi k/2)/(2N)}/2$ 的对应项相乘得到 $C(k)$。

例 10-9 利用MATLAB计算DCT

利用MATLAB计算信号 $x[n]$ 的DCT

$$x[n]=\sin\left(\frac{\pi n}{2N}+\frac{\pi}{6}\right)\quad N=8,n=0,1,2,\cdots,7$$

利用上述步骤通过DFT计算DCT，然后利用MATLAB函数dct检验所得结果。

解：先构建序列 $x[n]$，$n=0$，1，2，…，7，程序如下：

```
N = 8;
k = 0:N-1;
x = sin(2*pi*k/(4*N)+pi/6);
```

接着将 $x[n]$ 翻转得到 $xr[n]$，将其附加在 $x[n]$ 之后，得到信号 $xp[n]$。

```
for i = 1:N
    xr(i) = x(N-i+1);
end
xp = [x xr];
```

信号 $x[n]$ 和 $xp[n]$ 分别如图10.17a和图10.18所示。计算 $xp[n]$ 的DFT：

```
XP = fft(xp);
```

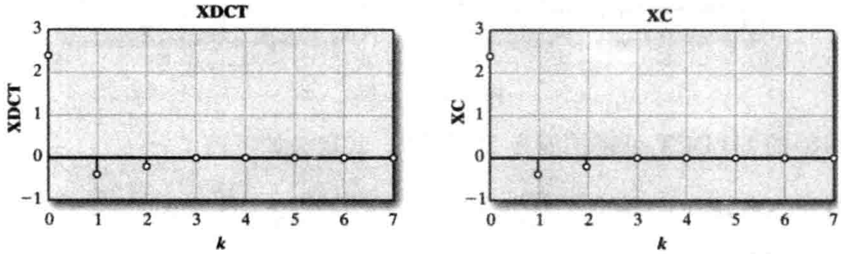
注意，$xp[n]$ 和 $XP[k]$ 的长度均为 $2N$。

为了从DFT中计算得到DCT，需将 $XP[k]$ 的每一项乘以 $p(k)\mathrm{e}^{\mathrm{j}(2\pi k/2)/(2N)}/2$，其中

$k=0$时，$p(k)=\sqrt{1/8}$；$k=1, 2, \cdots, 15$ 时，$p(k)=\sqrt{2/8}$，MATLAB 程序为：

```
W = exp(-j*2*pi*k/(4*N));
XC = W.*XP;
C1 = (1/N)^.5;
C2 = (2/N)^.5;
XC(1) = XC(1)C1/2;
XC(2:N) = XC(2:N)/C2/2;
```

MATLAB 程序中信号 XC 表示 $x[n]$的 DCT。为验证结果的正确性，同时采用 MATLAB 中的 dct 函数进行计算，两者结果比较见图 10.19 所示。



a）使用MATLAB中dct函数计算所得$x[n]$的DCT　　b）使用DFT计算得到的$x[n]$的DCT

图　10.19

```
XDCT = dct(x);
figure(1);clf;
k = 0:7;
subplot(1, 2, 1);
stem(k, XDCT);
subplot(1, 2, 2);
stem(k, XC(1:8));
```

◀

1D DCT 的基函数可以通过绘制式(10.10)中 DCT 中的余弦函数得到，其中，k 的范围为$[0, N-1]$。令 $N=16$，可以画出 16 个基函数图，如图 10.20 所示。

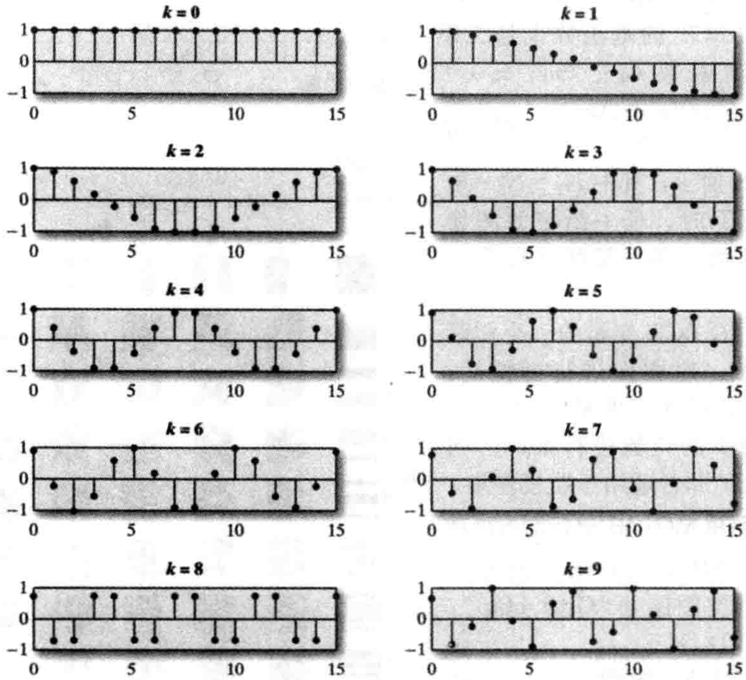


图 10.20　$N=16$ 时 1D DCT 的基函数图，基函数通过式 $C_k(n)=\cos(\pi k(2n+1)/(2N))$绘制

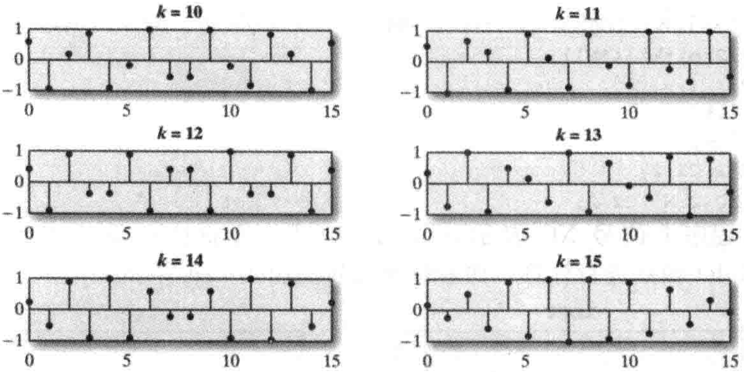


图 10.20 （续）

式(10.10)的 1D DCT 可以扩展为 2D DCT，其定义如下：

$$\textbf{2D DCT}\quad C(k_1,k_2)=pq\sum_{n_1=0}^{N_1-1}\sum_{n_2=0}^{N_2-1}x[n_1,n_2]\cos\frac{\pi(2n_1+1)k_1}{2N_1}\cos\frac{\pi(2n_2+1)k_2}{2N_2}\tag{10.11}$$

$$\textbf{2D IDCT}\quad x[n_1,n_2]=pq\sum_{k_1=0}^{N_1-1}\sum_{k_2=0}^{N_2-1}C(k_1,k_2)\cos\frac{\pi(2n_1+1)k_1}{2N_1}\cos\frac{\pi(2n_2+1)k_2}{2N_2}\tag{10.12}$$

式中，

$$p=\begin{cases}1/\sqrt{N_1}, & k_1=0\\ 2/\sqrt{N_1}, & 1\leqslant k_1\leqslant N_1-1\end{cases}\qquad q=\begin{cases}1/\sqrt{N_2}, & k_2=0\\ 2/\sqrt{N_2}, & 1\leqslant k_2\leqslant N_2-1\end{cases}$$

2D DCT 的性质如表 10.3 所示，其中，x 和 y 表示两个二维信号，在$[0, N_1-1]$范围外的值为零，其对应的二维离散余弦变换为 X 和 Y，即 $x\Leftrightarrow X$，$y\Leftrightarrow Y$。

表 10.3　2D DCT 的性质

线性	$ax+by\Leftrightarrow aX+bY$，其中 a、b 为常数
可分性	如果 x 可写成两个序列的乘积，则 x 的 DCT 为两个序列 DCT 的乘积，即，$x=x_1[n_1]\cdot x_2[n_2]\Leftrightarrow X_1(k_1)\cdot X_2(k_2)$
信号能量	$\sum_{n_1=0}^{N_1-1}\sum_{n_2=0}^{N_2-1}\lvert x\rvert^2=\frac{1}{pqN_1N_2}\sum_{k_1=0}^{N_1-1}\sum_{k_2=0}^{N_2-1}\lvert X\rvert^2$
对称性	(a) $x^*[n_1, n_2]\Leftrightarrow X^*(k_1, k_2)$ (b) $\mathrm{Re}(x[n_1, n_2])\Leftrightarrow\mathrm{Re}(X(k_1, k_2))$

基函数。2D DCT 的基函数也是二维函数，通过绘制函数 $C_{k_1,k_2}(n_1, n_2)=\cos\frac{\pi(2n_1+1)k_1}{2N_1}\cos\frac{\pi(2n_2+1)k_2}{2N_2}$ 可以得到基函数，如图 10.21 所示，其中 $N_1=N_2=8$。

使用 2D DCT 时，任一给定的图像表示为基函数的加权和。某些加权系数可能为零。

能量聚集。2D DCT 变换的一个实际应用是图像压缩。对图像进行变换后，可以将其能量重新排列为更紧凑的形式，从而实现采用较少的位数进行编码。当图像相关时，DCT 非常适用于能量聚集。

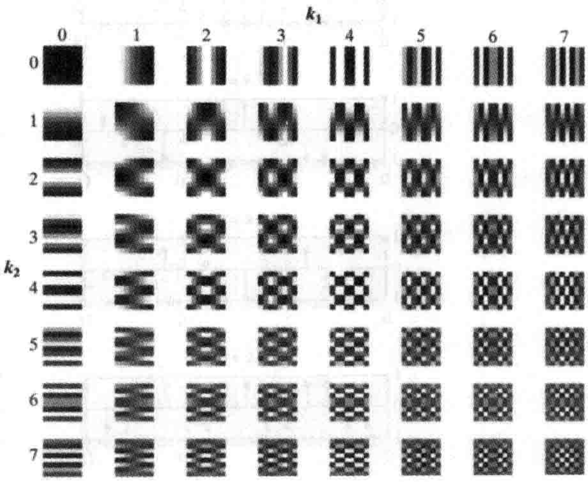


图 10.21　$N_1=N_2=8$ 时 2D DCT 的基函数

例 10-10　利用 MATLAB 进行图像处理

利用 MATLAB 提供的图像处理工具箱，绘制两个位图图像“Grid. bmp”、“Lines. bmp”的 2D DCT。

解：在 MATLAB 图像处理工具箱中，函数 image read 和 image write 可读

取和写入常见格式的图像，如位图图像。下面的程序实现读取图像以及对图像进行 DCT。

```
clear;
s = imread('Grid.bmp');
%Convert from color to 2D B & W
N1 = size(s, 1);N2 = size(s, 2);
z1 = zeros(size(s, 1),size(s, 2));
z1(1:N1, 1:N2) = s(1:N1,1:N2, 1);
figure(1);
imshow(z1, colormap(gray));
%
XDCT = dct2(z1);
%normalize 0 to 1 and reverse colors
mx = max(max(abs(XDCT)));
mn = min(min(abs(XDCT)));
XDCT = (XDCT+mn)/(mx - mn);
figure(2);clf;
z2 = 1 - abs(XDCT);
imshow(z2, [.99 1]);
```

在此例中，由于两幅图像的频率成分都相对较少，因而对 2D DCT 进行归一化后，采用彩色突出这些频率分量。其中阴影越深表示频率越高。程序运行的结果如图 10.22 所示。

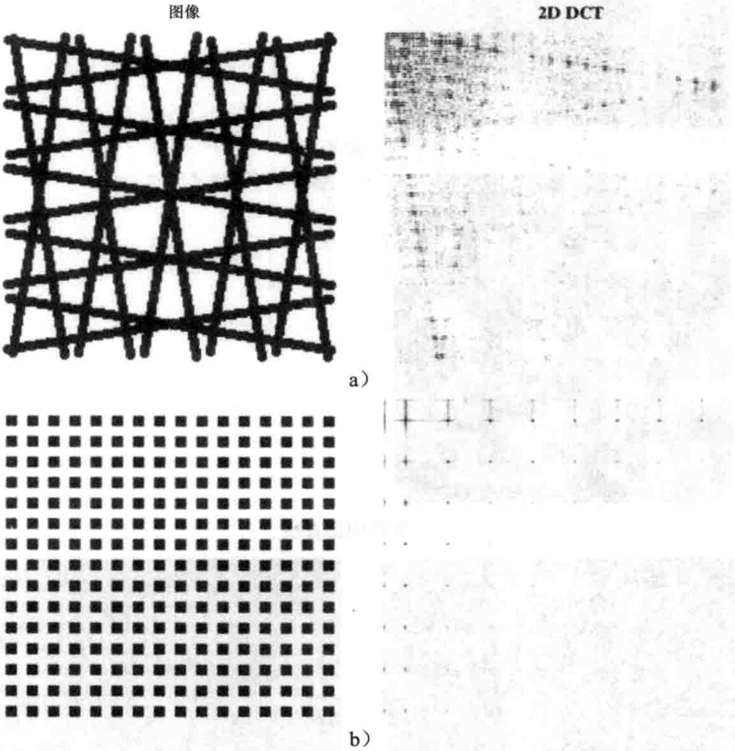


图 10.22　人工生成的图像及其 2D DCT，变换后图像的(0,0)点位于图像的左上角

图 10.23 为一些其他图像及其变换的举例。图 a～d 的频率成分逐渐增加，图 a 中，直流分量是最主要的频率成分，没有非常突出的单个频率，整体频率都比较低。图 b 中，高频分量来自于背景中的草地，但整个黑色部分色调很暗，白色部分很白且范围较大、对比度小，导致整体频率较低。图 c 中，图像是规则的方形图案，从而频域中水平和垂直方向的频率分布也较规则。然而，图像中方块之间的阴影是随意分布的，从而产生了一些高

频分量。图 d 中，树上的冰使得图像呈现明显的明暗对比，产生了较多的高频分量。这幅图像相关性小，能量紧缩率有限。

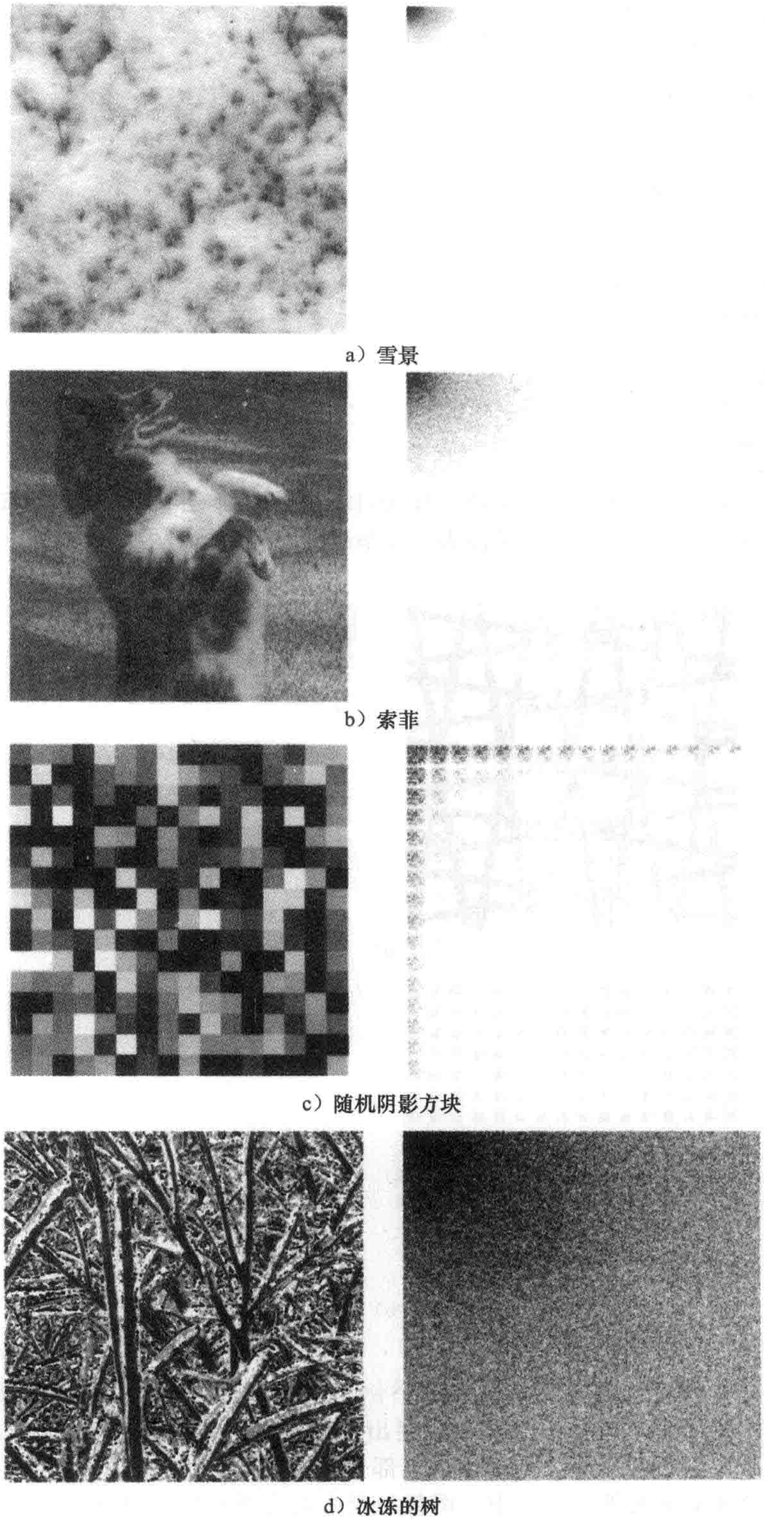


图 10.23　样本图像及其 2D DCT

10.3　二维 FIR 滤波器

一维 FIR 滤波器本质上是稳定的，在相同的参数指标下，其计算效率低于 IIR 滤波器，但其可实现线性相位，这些特性在二维情况下通常也成立。FIR 滤波器有限长的特点能确保脉冲响应绝对值的求和是有限的，因而在设计 FIR 滤波器时无须考虑稳定性问题。但是，二维 FIR 滤波器比 IIR 滤波器需要更多的乘法运算，因而需要占用更多的存储空间。与一维情况类似，二维 FIR 滤波器可以得到线性相位(或者零相位)，这使得在许多应用中，尽管存在计算上的差异，但 FIR 滤波器比 IIR 滤波器更理想。

回顾以前的知识，可知数字滤波器可具有零相位、线性相位或非线性相位。零相位滤波器的脉冲响应是一个偶函数，因此是非因果的。在一维情况下，通常先构建非因果脉冲响应的理想滤波器，然后将脉冲响应在时域平移再截断，使其变成因果且有限长的序列，从而能够实时实现。在许多应用中，数据可以进行存储，在数据处理时可以同时获取过去和将来的数据，所以不用考虑因果性的问题，这样的滤波器往往更高效，其在图像处理中得到应用。在图像滤波中，滤波器的脉冲响应表示为 $h[n_1, n_2]$，n_1、n_2 是空间变量，与时间无关。滤波器脉冲响应可以是关于原点对称且相位为零的形式，即 $h[n_1, n_2]=-h[n_1, n_2]$。对于二维情况，本书只介绍零相位的 FIR 滤波器。

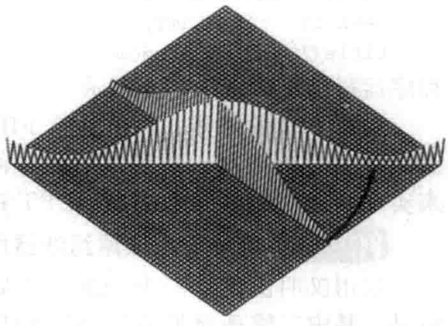


图 10.24　利用一维汉明窗函数围绕 x-y 平面旋转构造一个二维汉明窗

10.3.1　窗函数法

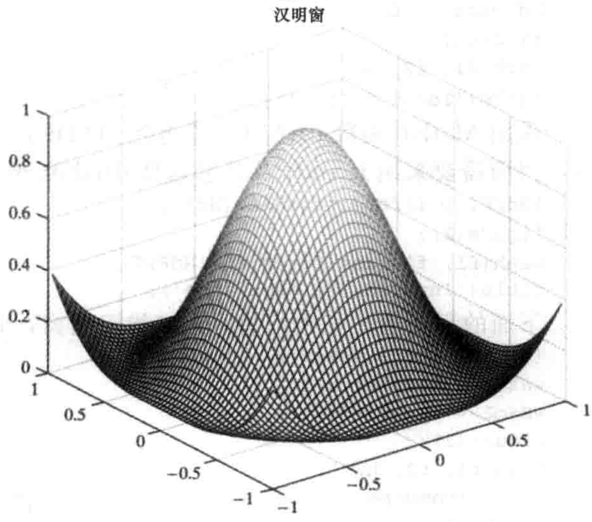
在一维情况下，窗函数法先构建一个理想滤波器，然后采用 IDFT 求出理想滤波器的脉冲响应，再将脉冲响应与在时域定义的窗函数相乘，从而获得加窗滤波器的脉冲响应。二维情况下的处理过程基本相同，只是用到的是二维窗函数。

二维窗函数可用多种方式构建，最常用的两种方式是旋转法和相乘法[5]。旋转法是通过构建一个一维窗函数，在二维平面上进行旋转，从而得到二维窗函数。相乘法是指用两个一维窗函数相乘得到二维窗函数，其原理与两个一维滤波器相乘得到二维滤波器的原理类似。

旋转窗。如图 10.24 所示，汉明窗函数围绕 x-y 平面旋转构造成一个"汉明表面"，又称为二维汉明窗。为了实现旋转，必须首先构建一个一维连续时间汉明窗，从而对于时间轴上的任一点都有取值。对于 x-y 平面上的某一点，其到中心点的距离为 r，旋转窗在该点的值等于一维汉明窗中距离中心点为 r 的点的取值。图 10.25 给出了该实现过

```
%TwoDHamming.m
N = 65;
[t1 t2] = freqspace(N, 'meshgrid');
r = sqrt(t1.^2 + t2.^2);
Wham = .54 + .46*cos(pi*sqrt(t1.^2 + t2.^2));
mesh(t1, t2, Wham);
```

a)



b)

图 10.25　旋转法构造二维汉明窗的 MATLAB 代码实现。其中四个角上的"尾巴"由于对角长度大于窗中心到水平和垂直边缘的长度而造成

程的 MATLAB 代码。

乘法窗。将一个一维窗函数取值写成行向量，另一窗函数取值写为列向量，两者相乘即可得到二维窗函数。通常这两个向量用同一个函数表示，尽管算法没有对此作要求。对列向量和行向量做矩阵乘法即可得到二维窗函数，MATLAB 代码如下。

```
N = 65;
[t1 t2] = freqspace(N, 'meshgrid');
wHam1 = hamming(N);
wHam2 = wHam1*wHam1';
mesh(t1, t2, wHam2);
title('Hamming window');
```

程序运行结果如图 10.26 所示。

加窗 FIR 滤波器。对于加窗 FIR 滤波器，二维窗函数通过逐项与理想滤波器的脉冲响应函数相乘来实现。脉冲响应可通过 2D IDFT 获得。◀

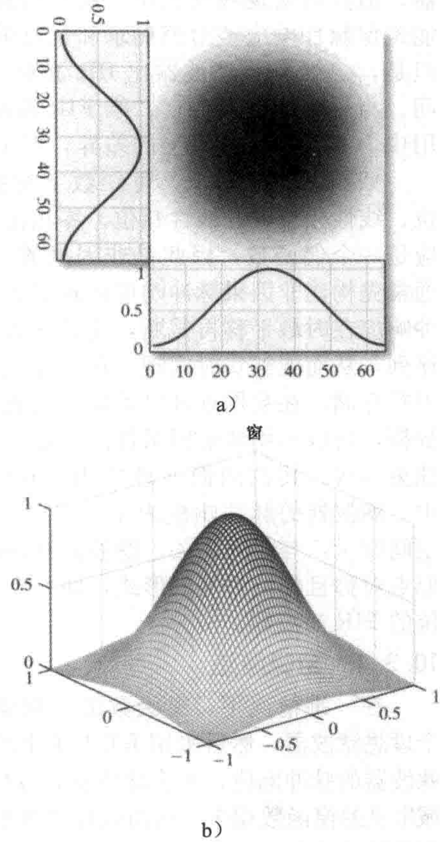


图 10.26　两个一维汉明窗相乘得到的二维汉明窗

例 10-11　二维理想低通滤波器的设计

试用汉明窗设计一个二维低通滤波器，并显示结果。其中二维理想低通滤波器的归一化抽样频率为 2，截止频率为 0.3。

解：利用 Matlab 程序产生一个二维频率空间的矩阵来构建理想滤波器，人工设定矩阵中频率小于 0.3 的值为 1，其余值设定为 0。

```
fs = 2;
fc = .3;
N = 65;
[f1 f2] = freqspace(N, 'meshgrid');
r = sqrt(f1.^2 + f2.^2);
Hd = ones(N);
Hd(r>fc) = 0;
figure(1);
mesh(f1, f2, Hd);
title('Ideal filter');
```

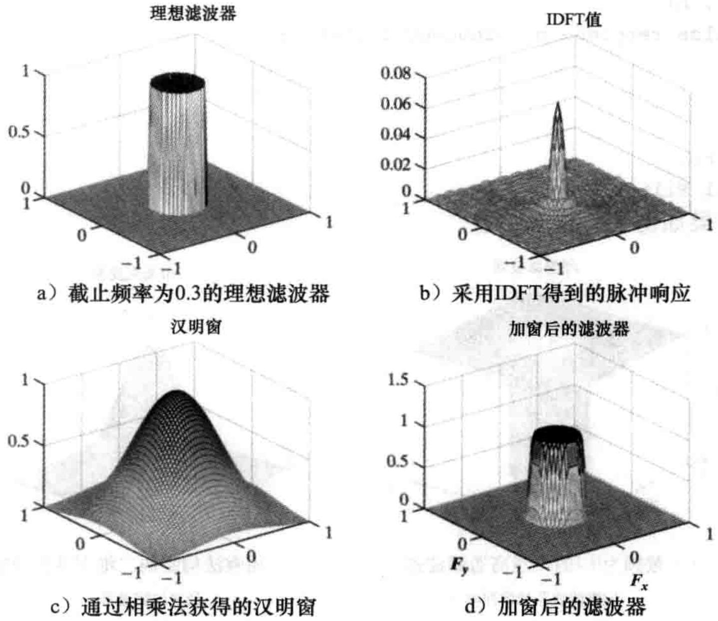
采用 Matlab 函数 ifft2 可计算得到 2D DFT。由于 fft2 和 ifft2 函数都默认起点在左下角，因而需要采用 fftshift 及其逆运算 ifftshift 平移起点。下面的程序实现 DFT 的反变换。

```
iHdFFT = ifft2(ifftshift(Hd));
figure(2);
mesh(f1, f2, abs(fftshift(iHdFFT)));
title('Inverse FFT magnitude');
```

下面的程序通过相乘构建一个二维汉明窗，并将其与理想滤波器的脉冲响应相乘。

```
[t1 t2] = freqspace(N, 'meshgrid');
wHam1 = hamming(N);
wHam2 = wHam1*wHam1';
figure(3);
mesh(t1, t2, hw2);
title('Hamming window');
figure(4);
freqz2(wHam2.*fftshift(iHdFFT));
title('Final filter');
```

程序运行结果如图 10.27 所示。



a）截止频率为0.3的理想滤波器　　b）采用IDFT得到的脉冲响应

c）通过相乘法获得的汉明窗　　d）加窗后的滤波器

图 10.27　例 10-11 图

利用窗函数设计时，可以使用任意一维窗函数。例如，凯瑟窗可以更有效控制过渡带宽度和波动。二维窗函数设计法的优点在于其相对简单。

MATLAB 中有两个函数可以自动实现窗函数法设计 FIR 滤波器。fwind1 函数利用一个一维窗构造二维 FIR 滤波器。该函数利用频率抽样法获得理想的频率响应。fwind2 函数采用用户自定义的窗函数设计二维 FIR 滤波器，脉冲响应由 IDFT 计算得到。

例 10-12　窗函数法构造二维 FIR 高通滤波器

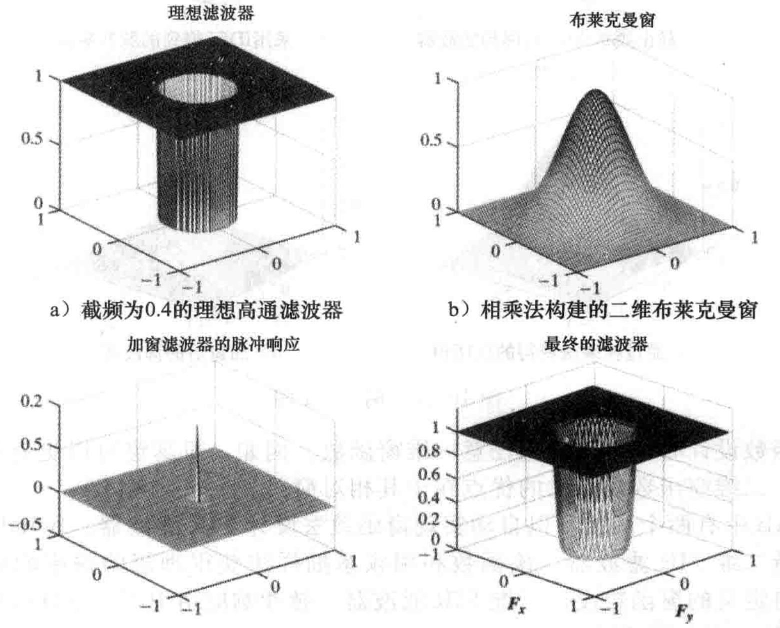
利用 MATLAB 提供的 fwind2 函数，试用二维布莱克曼窗设计一个二维 FIR 高通滤波器，滤波器的归一化抽样频率为 2.0，截止频率为 0.4。

解：

```
%BlackmanExmp.m
fs = 2;fc = .4;N = 65;
[f1 f2] = freqspace(N, 'meshgrid');
r = sqrt(f1.^2 + f2.^2);
Hd = ones(N);
Hd(r<fc) = 0; %Ideal filter
figure(1);
mesh(f1, f2, Hd);
title('Ideal filter');
%
wBlack1 = blackman(N);
wBlack2 = wBlack1*wBlack1'; %2D by multiplication
figure(2);
mesh(f1,f2, wBlack2);
title('Blackman window');
%
h = fwind2(Hd, wBlack2);
figure(3);
```

```
mesh(f1, f2, h);
title('Impulse response of windowed filter');
%
figure(4);
freqz2(h);
H = freqz2(h);
title('Final Filter');
```

程序运行结果如图 10.28 所示。



a）截频为0.4的理想高通滤波器　b）相乘法构建的二维布莱克曼窗

c）设计的滤波器的脉冲响应　d）设计的滤波器的频率响应

图 10.28 例 10-12 图

10.3.2 二维频率抽样

在频率抽样法中，设计者在频域内设定理想滤波器的有限抽样点。在一维情况下，可以获得一个通过所有抽样点的函数，可以把它看作对抽样点进行曲线拟合以逼近理想滤波器。一维函数的性质见表 5.6。在二维情况下，也可以利用同样的方法进行频率抽样，但得到的是一个通过所有抽样点的平面，而不是曲线。

使用频率抽样法时，先定义一个理想二维滤波器，对其求 IDFT。由于 DFT 仅在第一象限内有定义，因此需将脉冲响应平移才能得到零相位的滤波器。

例 10-13 利用频率抽样法设计低通滤波器

利用频率抽样法设计一个低通滤波器，归一化抽样频率为 2，截止频率为 0.3。

解： 利用 MATLAB 构建理想滤波器

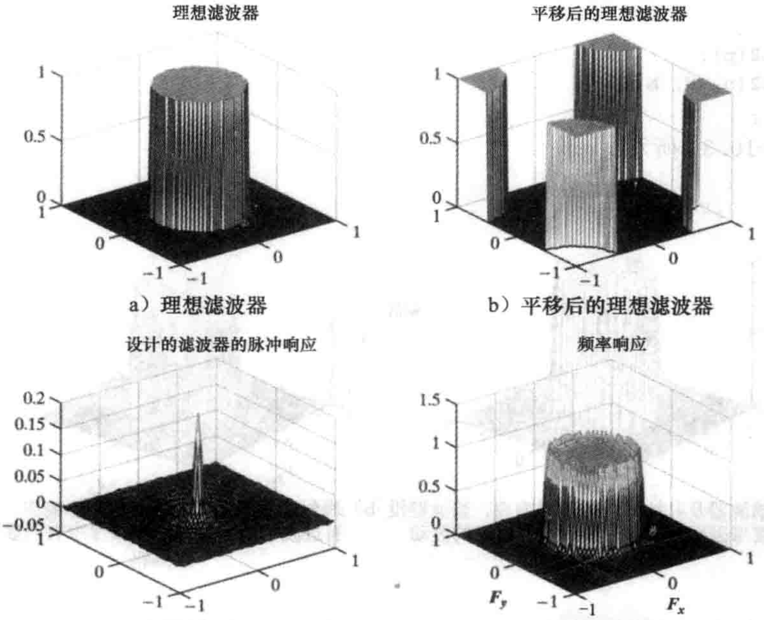
```
N = 65;
f = 2;
fc = 0.3;
p = ones(N);
[f1 f2] = freqspace(N, 'meshgrid');
r = sqrt(f1.^2 + f2.^2);
p(r > fc) = 0;
```

调用 fftshift 函数及其逆运算 ifftshift 将滤波器移至第一象限，再进行 IDFT，再次平

移滤波器使其相位为零，得到频率响应。

```
h = ifft2(ifftshift(p));
h = fftshift(h);
figure(1);
freqz2(h);
```

程序运行结果如图 10.29 所示。



a）理想滤波器　b）平移后的理想滤波器

c）设计的滤波器的脉冲响应　d）设计的滤波器的频率响应

图 10.29　例 10-13 图

MATLAB 函数 fsamp2 可自动实现上述抽样过程。◀

如果理想滤波器的截止边缘十分陡峭，那么频率抽样法拟合的所有抽样点的平面效果不好，会产生明显的波动。若要弥补此不足，可以通过给理想滤波器增加一个过渡带实现，具体过程见下例。

例 10-14　比较 FIR 滤波器的频率响应

利用频率抽样法设计两个 FIR 滤波器，一个截止频率为 0.5 且没有过渡带，另一个滤波器有过渡带且通带截止频率为 0.5，阻带截止频率为 0.57，两个滤波器的抽样频率均为 2，比较这两个滤波器的频率响应。上述频率皆为归一化频率。

解：利用 MATLAB 设计第一个滤波器：

```
N = 33;
fps = 0.5;
p = ones(N);
[f1 f2] = freqspace(N, 'meshgrid');
r = sqrt(f1.^2 + f2.^2);
p(r > fps) = 0;
```

对于第二个滤波器，过渡带是一条从点(fps，1)到点(fst，0)的直线，fps 和 fst 分别表示通带截止频率和阻带截止频率。过渡带定义为

$$y=-\frac{1}{\mathrm{fst}-\mathrm{fps}}x+\frac{\mathrm{fst}}{\mathrm{fst}-\mathrm{fps}}$$

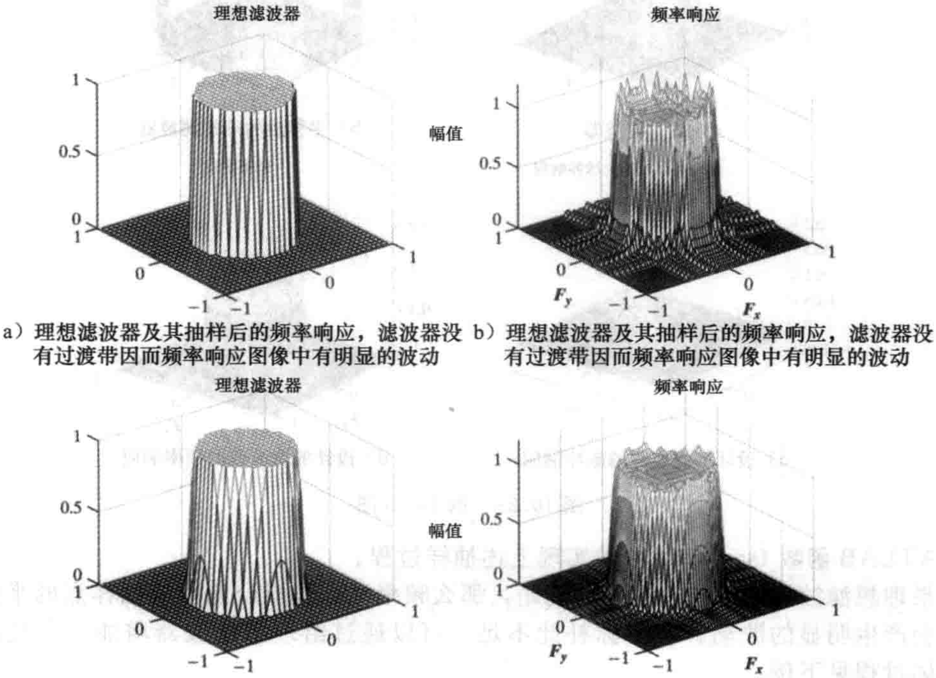
利用 MATLAB 设计一个具有过渡带的二维滤波器，程序如下：

```
N = 33;
fps = 0.5;
fst = 0.57;
p = ones(N);
p(r > fps) = -(1/(fst-fps))*r(r > fps) + fst/(fst - fps);
p(r > fst) = 0;
```

对于这两个滤波器也可用 fsamp2 函数实现频率抽样，用 freqz2 函数求解它们的频率响应：

```
h = fsamp2(p);
H = freqz2(h, N, N);
freqz2(h);
```

结果如图 10.30 所示。



a）理想滤波器及其抽样后的频率响应，滤波器没有过渡带因而频率响应图像中有明显的波动 b）理想滤波器及其抽样后的频率响应，滤波器没有过渡带因而频率响应图像中有明显的波动

c）滤波器具有过渡带，波动明显减少 d）滤波器具有过渡带，波动明显减少

图 10.30 例 10-14 图 ◀

10.3.3 变换法

利用变换法设计二维滤波器是指将一维滤波器经数学变换，使其成为二维滤波器，这个过程类似于低通滤波器进行频率变换后得到带通滤波器。变换法是由 McClellan 提出的，最初是将一维数字滤波器写成余弦形式。如果一个 FIR 滤波器的脉冲响应是对称的，$h[n]=h[-n]$，则频率响应可写为

$$H(e^{j\omega T}) = \sum_{n=-(N-1)/2}^{(N-1)/2} h[n]e^{-j\omega T} = \sum_{n=0}^{(N-1)/2} k[n]\cos(\omega nT) \tag{10.13}$$

式中，

$$k[n] = \begin{cases} h[0] & n=0 \\ 2h[n] & \text{其他} \end{cases}$$

可以看出 $k[n]$ 是常数，方程中的余弦是 $\cos(nx)$ 的形式。联想到 5.1 节，切比雪夫多项式可以写为

$$T_N(x) = \cos(N\arccos(x))$$

令 $x=\cos(\Omega)$，得到

$$T_0(\cos(\Omega)) = 1$$

$$T_1(\cos(\Omega)) = \cos(\Omega)$$

为了得到更多项，可利用递归关系式

$$T_{k+1}(\cos(\Omega)) = 2\cos(\Omega)T_k(\cos(\Omega)) - T_{k-1}(\cos(\Omega)) \tag{10.14}$$

如果展开多项式至高阶，$\cos(n\Omega)$可以表示成 $\cos^n(\Omega)$的线性组合，即

$$\cos(n\Omega) = \sum_{k=0}^{n} b_k \cos^k(\Omega) \tag{10.15}$$

式中，b_k 为常数。

将式(10.15)代入式(10.13)，可得

$$H(e^{j\Omega}) = \sum_{n=0}^{(N-1)/2} a_n \cos^n(\Omega) \tag{10.16}$$

最后，利用变换公式将式(10.16)从一维滤波器的频率响应变成一个二维信号。在 McClellan 的原著中，所使用的变换式为

$$\cos(\Omega) = T(\Omega_1, \Omega_2)$$

式中，$\Omega=\sqrt{\Omega_1^2+\Omega_2^2}$且

$$T(\Omega_1, \Omega_2) = 0.5[\cos(\Omega_1) + \cos(\Omega_2) + \cos(\Omega_1)\cos(\Omega_2) - 1] \tag{10.17}$$

例 10-15 将一维滤波器变换成二维滤波器

利用变换法构造一个二维滤波器，给定的一维滤波器为

$$H(z) = \frac{z^2 + 1.5z + 0.5}{z^2}$$

解：一维滤波器的脉冲响应为

$$h[n] = \{\hat{1}, 1.5, 0.5\}$$

符号$\hat{}$指向 $h[0]$。若将滤波器变为零相位，则有

$$h[n] = \{0.5, 1.5, \hat{1.0}, 1.5, 0.5\}$$

相应的 z 变换为

$$H(z) = 0.5z^2 + 1.5z + 1.0 + 1.5z^{-1} + 0.5z^{-2} \tag{10.18}$$

可写为

$$H(z) = \frac{z^2 + z^{-2}}{2} + \frac{3(z + z^{-1})}{2} + 1$$

用 $e^{j\omega t}$代替 z，得到频率响应

$$H(e^{j\omega T}) = \cos(2\omega T) + 3\cos(\omega T) + 1$$

由切比雪夫递归关系式(10.14)得

$$\cos(2\omega T) = 2\cos^2(\omega T) - 1$$

从而得到频率响应

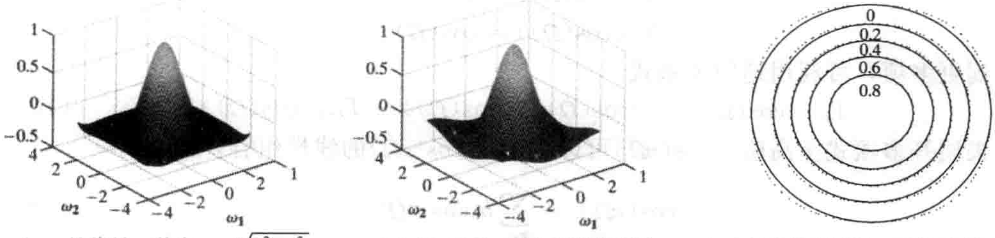
$$H(e^{j\omega T}) = 2\cos^2(\omega T) + 3\cos(\omega T)$$

利用 McClellan 的变换式(10.17)，经过大量代数运算，得到

$$\begin{aligned} H(e^{j\omega T}) = & 0.5[\cos^2(\omega_1 T) + \cos^2(\omega_2 T)] + 2\cos(\omega_1 T)\cos(\omega_2 T) \\ & + \cos^2(\omega_1 T)\cos(\omega_2 T) + \cos(\omega_1 T)\cos^2(\omega_2 T) \\ & + 0.5[\cos(\omega_1 T)\cos(\omega_2 T)] + 0.5[\cos^2(\omega_1 T)\cos^2(\omega_2 T)] - 1 \end{aligned}$$

利用 MATLAB 绘制了在 ω_1、ω_2 平面内的幅频响应图。为了进行比较，也绘制出原

滤波器的幅频响应图，即 $H(e^{j\omega T})=\cos(2\omega T)+3\cos(\omega T)+1$，式中，$\omega=\sqrt{\omega_1^2+\omega_2^2}$。程序运行结果以及等高线图如图 10.31 所示。



a）一维信号，其中$\omega=\sqrt{\omega_1^2+\omega_2^2}$　b）McClellan变换的近似结果　c）合并的等高线图，虚线为近似结果

图 10.31　例 10-15 图

同样，可以通过 z 变换得到例 10-15 的结果。为了得到频率响应，利用 $e^{j\omega T}$ 代替 z，则 $\cos(\omega T)=(e^{j\omega T}+e^{-j\omega T})/2$ 与 $(z+z^{-1})/2$ 对应，将此替换应用到 McClellan 变换中

$$T(z_1,z_2)=0.5\left(\frac{z_1^2+1}{2z_1}+\frac{z_2^2+1}{2z_2}+\frac{z_1^2+1}{2z_1}\frac{z_2^2+1}{2z_2}-1\right)$$

展开此方程，可以得到如图 10.32 所示的二维滤波器掩码。

将 $z\leftarrow T(z_1,z_2)$代入式(10.18)，得到的频率响应与例 10-15 的结果相同。

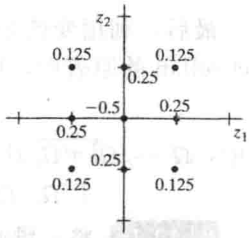


图 10.32　McClellan 变换得到的二维滤波器掩码

MATLAB 提供的 ftrans2 函数可自动完成二维变换的设计过程。调用该函数时，首先要利用常规的设计方法设计一个一维滤波器，满足滤波器的长度为奇数(偶数阶)，再调用 ftrans2 函数构造二维滤波器。

例 10-16　利用 MATLAB 转换一维低通滤波器为二维低通滤波器

调用 firls 函数设计一个一维低通滤波器，归一化抽样频率为 2，过渡带频率范围为[0.3，0.35]，滤波器阶数为 16，调用函数 ftrans2 函数将其转换成二维滤波器。

解：利用 MATLAB 设计一维滤波器：

```
fs = 2;
F = [0 .3 .35 1];
M = [1 1 0 0];
N = 16;
b = firls(N, F, M);
[H f] = freqz(b, 1, 1024, fs);
figure(1);
plot(f, abs(H))
```

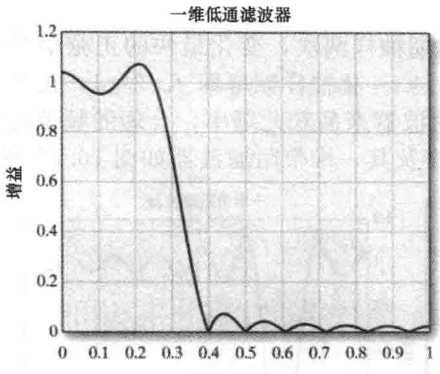
调用 ftrans2 函数构造二维滤波器并绘制其频率响应图：

```
h = ftrans2(b);
figure(2);
[H f1 f2] = freqz2(h, [129 129]);
mesh(f1, f2, abs(H));
```

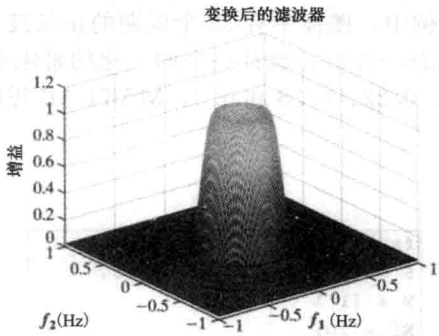
为得到二维滤波器的横截面，编写以下代码：

```
cross = abs(H(:,64));
plot(f1, (cross));
```

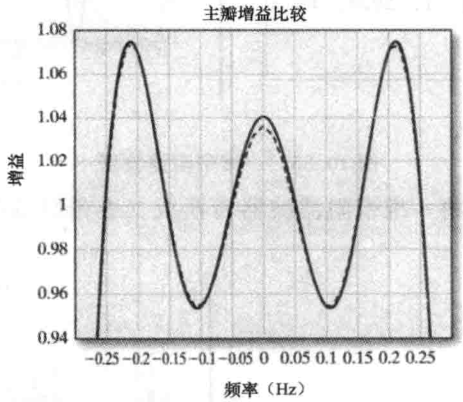
结果如图 10.33 所示。二维滤波器的横截面图用于与一维滤波器的幅度图比较。



a）原始一维滤波器的幅度响应图



b）经过变换后的二维滤波器的幅度响应图



c）二维滤波器幅度响应的主瓣横截面图，叠加的曲线为一维滤波器的幅度响应

图 10.33　例 10-16 图

10.3.4　FIR 滤波器在图像中的应用

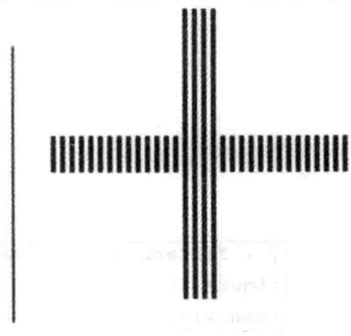
MATLAB 提供的 filter2 函数可用于图像滤波，其自变量为输入图像和给定的 FIR 滤波器，输出为滤波后的图像。下面的例子介绍如何调用该函数验证滤波器设计。

例 10-17　利用 MATLAB 设计一个 30 阶的 FIR 带阻滤波器，并将其转换成二维滤波器

下面的程序运行后产生一个大加号“+”，并以正弦波纹填充，如图 10.34 所示。程序中调用 firls 函数构造 30 阶的 FIR 带阻滤波器，将其转换成二维滤波器，然后用这个滤波

```
N = 256;
z1 = ones(N);
n2 = 16;
f = 32;
for i = 1:N
    z1(:, i) = (1+sin(2*pi*f*i/N))/2;
end
z1(1:N/2-n2, 1:N/2-n2) = 1;
z1(1:N/2-n2, N/2+n2:N) = 1;
z1(N/2+n2:N, 1:N/2-n2) = 1;
z1(N/2+n2:N, N/2+n2:N) = 1;
```

a）



b）

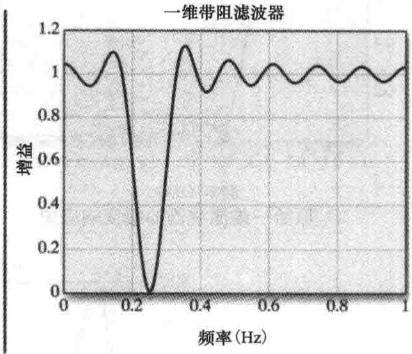
图 10.34　MATLAB 生成的二维图像

器过滤图像中的波纹。

解： 图像由256个抽样方块构成，每个周期抽样两次，变化最快的正弦信号有128个周期。此例中，图像中有32个周期的正弦波，归一化抽样频率即 $f_s/2=1$，正弦波对应频率为32/128＝0.25。设计一个归一化的带阻滤波器来截获此频率，合理的频带边界频率可设为0.2、0.22、0.28和0.3。MATLAB程序及其一维带阻滤波器如图10.35所示。

```
fs = 2;
F = [0 .20 .22 .28 .3 1];
M = [1 1 0 0 1 1];
Nf = 30;
b = firls(Nf, F, M);
[H f] = freqz(b, 1, 1024, fs);
figure(1);
plot(f, abs(H));
```

a)



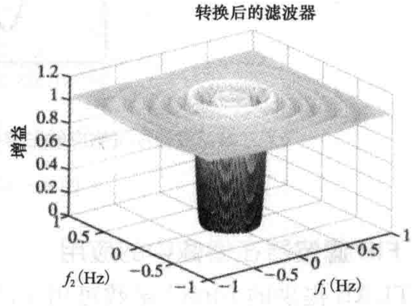
b)

图10.35 一维带阻滤波器

下面调用ftrans2函数将一维带阻滤波器转换成二维带阻滤波器，所得滤波器的响应如图10.36所示。

```
h = ftrans2(b);
figure(1);
[H f1 f2] = freqz2(h, [128 128]);
mesh(f1, f2, abs(H));
```

a)



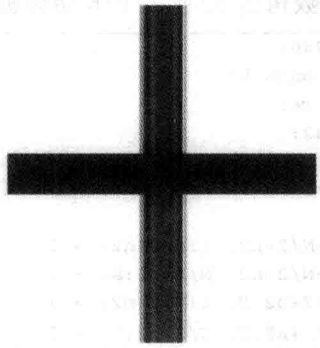
b)

图10.36 由图10.35中一维滤波器转换得到的二维滤波器

最后，调用filter2函数实现用该二维滤波器对图像的滤波，结果如图10.37所示。由

```
y = filter2(b, z1, 'valid');
figure(1);
imshow(y, [.1 .9]);
```

a)



b)

图10.37 滤波后的图像

图 10.37 可见，水平横条上的正弦频率几乎全被去除，而竖条的左右边缘附近留有其他频率，这些是正弦信号被截断时产生的较高频率。◀

例 10-18 利用 MATLAB 实现对 Sophie 图像的滤波

图 10.38 的图像名为"Sophie"，借助 fir1 函数构建 32 阶的高通滤波器，对图像进行滤波。

解： 下面的 MATLAB 程序实现读取图像，构造二维滤波器，以及调用 filter2 函数对图像进行滤波。图 10.39 显示了不同截止频率下的滤波结果。

```
s = imread('sophieyaya.bmp');
z = rgb2gray(s);
figure(1);clf;
imshow(z);
%
Nf = 32;
fs = 2;
figure(2);clf;
fc = .05;
b = fir1(Nf, fc(i), 'high');
[H f] = freqz(b, 1, 1024, fs);
h = ftrans2(b);
y = filter2(h, z);
imshow(y);
```



图 10.38 Sophie 图像

图 10.39a 中截止频率相对较低，所去掉的低频分量很少，图像中大片的白色区域得以保留。相比较而言，图 10.39d 中去掉了较多低频分量，仅有变化比较明显的区域被保留下来。

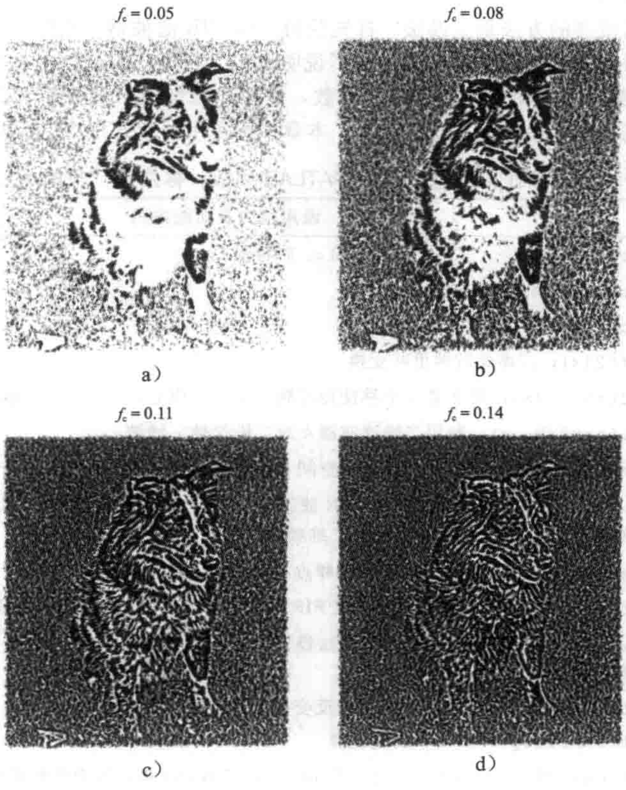


图 10.39 图像"Sophie"分别通过 4 个不同截止频率的高通滤波器。图 a～d 中频率由 0.05 逐步增加到 0.14，步长为 0.03 ◀

小结

人们提到信号的频率，往往认为是听觉上的感受。如果从视觉上分析频率，通常是指物体的颜色，因为不同频率的可见光代表不同的颜色。在本章中，可以看到从视觉上分析图像时，频率分量可表示图像中相邻像素间亮度的变化率。明亮的区域在极短的空间范围过渡到较暗的区域，表示存在高频分量的边缘，就像方波陡峭的边缘对应电压信号的高频分量。

图像可以用二维信号表示，同时大部分应用于一维数字信号处理的技术可以应用到二维，但有一些差别需引起注意。这里的信号不再是传统观念上随时间变化的电压信号，而是指在空间中变化的亮度或幅度值。在二维数字信号处理中，因果性也不再那么重要。二维信号的频率变化以及滤波器的稳定性很难甚至不可能以可视化表示(如零极点分布图)进行处理。

二维 DFT 可以通过多个一维 DFT 进行求解。分析一个 $N\times M$ 的图像，需对图像信号进行 M 次的 N 点 1D DFT，再进行 N 次 M 点的 1D DFT。在 2D DFT 中，通常将零频放在中间位置，并以该点为中心的同心圆表示其他较高频率。在傅里叶光学理论中，一个透镜在焦平面产生的效果相当于二维傅里叶变换，可以利用 2D DFT 模拟此等效过程。

DCT 是另一种图像变换方法，已广泛应用于图像压缩和编码。在压缩图像时，DCT 可用较少的位数对低能量图像进行编码，同时 DCT 可以从 DFT 计算得到。

利用窗函数法设计二维 FIR 滤波器时，实现过程与一维情况类似。本章介绍了两种生成二维窗函数的方法。第一种方法将传统的一维窗函数绕原点旋转获得二维窗函数平面。第二种方法用两个一维窗函数相乘，一个窗函数表示一个行向量，另一个窗函数(可能与第一个窗函数相同)则表示一个列向量。两个向量相乘所得矩阵即为二维窗函数。

频率抽样法设计一维 FIR 滤波器可扩展到二维，为此，我们需要对二维理想滤波器抽样后进行二维 IDFT。由于 DFT 定义在第一象限，需要利用对称性在其他三个象限构建类似的表面，从而得到二维 FIR 滤波器，其生成相对容易。

第三种设计二维滤波器的方法是变换法。首先获得一维 FIR 滤波器，根据一定的变换规则，将其变换成二维滤波器，并对周期对称 FIR 滤波器进行了说明。

MATLAB 的图像处理工具箱中提供了许多函数，使得二维情况下的分析和处理过程更为简单。可以使用 filter2 函数实现二维滤波并显示滤波结果。本章中所使用的 MATLAB 函数见表 10.4。

表 10.4　本章中用于滤波器设计和图像显示的 MATLAB 函数，标注“*”号的函数来自图像处理工具箱

函数	调用格式及功能描述
conv2	y= conv2(x1, x2)；二维信号 x_1 和 x_2 的卷积
Dct	y= dct(x)；离散余弦变换
dct2*	y= dct2(im)；二维离散余弦变换
fft2	y= fft2(x)；二维离散傅里叶变换
fftshift	y= fftshift(x)；将变量 x 平移使得零频分量位于中心，x 可以是一维或二维变量
filter2	y= filter2(h, x)；利用二维滤波器 h 对二维信号 x 滤波
freqspace	f1 f2]= freqspace(N)；构建频率空间，一般 f_1、f_2 的取值范围为[−1, 1]，以 N 步实现
freqz2*	[H f1 f2]= freqz2(h)；求二维 FIR 滤波器 h 对应的频率响应 H，并返回两个频率向量 f_1 和 f_2。单独使用函数 freqz2(h)可绘制二维频率响应图
fsamp2*	h= fsamp2(H)；利用 H 中的频率抽样点构建二维 FIR 滤波器
ftrans2*	h= ftrans2(b)；通过变换法将一维 FIR 滤波器 b 变换成二维 FIR 滤波器 h
fwind2*	h= fwind2(Hd, win)；利用窗函数法设计二维 FIR 滤波器。Hd 是理想滤波器的频率响应，win 是二维窗函数
ifft2	y= ifft2(x)；计算二维离散傅里叶反变换
ifftshiftP	x= ifftshift(y)；fftshift 的逆运算
imread	A= imread('filename', fmt)；从 filename 读取 fmt 格式的图像数据至二维向量 A 中，fmt 可为 jpeg、bmp 等格式
imshow*	imshow(im)；显示灰度图像

（续）

函数	调用格式及功能描述
imwrite	imwrite(A, 'filename', fmt)；将灰度图像 A 以 fmt 格式写入文件 filename
mesh	mesh(x, y, z)或 mesh(z)；生成由 x、y、z 定义的网格线表面图
surf	surf(x, y, z)或 surf(z)；生成三维曲面图，颜色变化与高度成比例

习题

思考题

1. 阐述离散余弦变换与离散傅里叶变换之间的关系。
2. 假如我们利用一维 FFT 计算一个 $N\times N$ 图像的二维 FFT，其中复数相乘的次数如何随 N 变化？
3. 二维傅里叶变换与光学中的干涉现象有什么关系？结合光干涉现象启发性解释竖向细缝的二维傅里叶变换。
4. DFT 可用于计算循环卷积，如何利用 DFT 计算线性卷积？
5. DCT 能量聚集的含义是什么？
6. 一维滤波系统中，水平轴往往是时间轴，并且常讨论系统的因果性；二维滤波系统中，两个维度都表示空间位置，且系统的因果性变得不重要，这是否意味着可以实现理想滤波器？
7. 给定一个一维滤波器 $h_1[n]\leftrightarrow H_1(z)$，旋转$h_1[n]$得到二维脉冲响应 $h_1[n_1,n_2]$。$h_1[n_1,n_2]$的 z 变换为 $H_1(z_1,z_2)$，证明 $H_1(z_1,z_2)$不可由 $H_1(z)$旋转得到。
8. 当计算一维信号的 FFT 时，可以得到图 10.8 的结果，其中抽样频率为 1，观察此图可得：
   - 频率从 0Hz 线性变化到 f_s。
   - 频率分辨率为 f_s/L，L 是信号抽样点的个数。
   - 图形是左右对称的。
   - FFT 的幅值以左侧的线性尺度标注。

   这些特性在二维 FFT 的频率显示结果中有哪些不同？

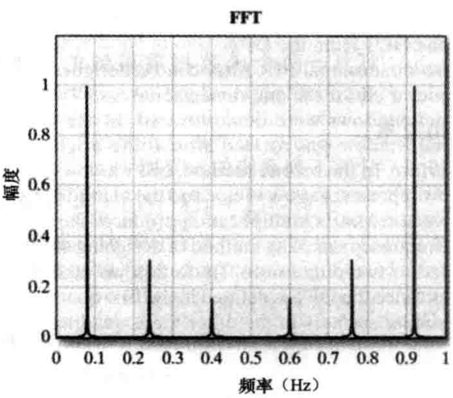


图 P10.8　思考题 8 的 FFT 图

分析和设计题

10.1 节

10.1　已知二维 FIR 滤波器的差分方程

$$y[k_1,k_2]=b_{0,0}x[k_1,k_2]-b_{0,1}y[k_1,k_2-1]-b_{1,0}y[k_1-1,k_2]-b_{1,1}y[k_1-1,k_2-1]$$

式中，$b_{0,0}=1$，$b_{0,1}=0.5$，$b_{1,0}=0.5$，$b_{1,1}=0.25$。

试确定滤波器的脉冲响应。

10.2　已知二维 IIR 滤波器的差分方程：

$$y[k_1,k_2]=b_{0,0}x[k_1,k_2]-a_{0,1}y[k_1,k_2-1]-a_{1,0}y[k_1-1,k_2]-a_{1,1}y[k_1-1,k_2-1]$$

式中，$b_{0,0}=1$，$a_{0,0}=1$，$a_{0,1}=0.5$，$a_{1,0}=-0.5$，$a_{1,1}=0.25$。

试确定滤波器的脉冲响应。

10.3　已知差分方程

$$y[k_1,k_2]=b_{0,0}x[k_1,k_2]-b_{0,1}y[k_1,k_2-1]-b_{1,0}y[k_1-1,k_2]-b_{1,1}y[k_1-1,k_2-1]$$

式中，$b_{0,0}=1$，$b_{0,1}=0.5$，$b_{1,0}=0.5$，$b_{1,1}=0.25$。

试画出输入/输出(I/O)掩码图。

10.4　已知差分方程

$$a_{0,0}y[k_1,k_2]=b_{0,0}x[k_1,k_2]+b_{1,1}x[k_1-1,k_2-1]-a_{0,1}y[k_1,k_2-1]-a_{1,0}y[k_1-1,k_2]-a_{1,1}y[k_1-1,k_2-1]$$

式中，$b_{0,0}=1$，$b_{1,1}=-1$，$a_{0,0}=-0.2$，$a_{0,1}=0.5$，$a_{1,0}=0.5$，$a_{1,1}=-0.25$。

试画出输入输出(I/O)掩码图。

10.5 根据下面绘制的掩码图，试确定相应的差分方程。

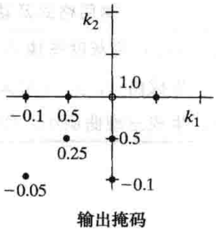


图 AP10.5 确定 I/O 掩码图所对应的差分方程

10.6 给定两个函数

$$f(k_1)=\begin{cases}0, & k_1<0\\ 1-k_1/10, & k_1\geqslant 0\end{cases} \quad 和 \quad g(k_2)=\begin{cases}0, & k_2<0\\ 1-k_2/10, & k_2\geqslant 0\end{cases}$$

计算这两个函数相乘得到的二维差分方程，并画出类似图 10.4 所示的结果图，网格大小为 10×10。

10.7 给定两个函数

$$f=g=\begin{cases}0, & k_1<0\\ e^{-k_1/40}\cos(2\pi f_1 k_1), & k_1\geqslant 0\end{cases}, \quad 式中，f_1=0.05$$

计算这两个函数相乘得到的二维差分方程，并画出类似图 10.4 所示的结果图，网格大小为100×100。

10.2 节

10.8 计算下列系统的 z 变换

$$\begin{aligned}y[k_1,k_2]=&x[k_1,k_2]+0.5x[k_1,k_2-1]+0.5x[k_1-1,k_2]+0.25x[k_1-1,k_2-1]\\&+0.25x[k_1-2,k_2]+0.25x[k_1,k_2-2]+0.25x[k_1-1,k_2-2]\\&+0.25x[k_1-2,k_2-1]+0.125x[k_1-2,k_2-2]\end{aligned}$$

10.9 计算下列系统的 z 变换

$$\begin{aligned}y[k_1,k_2]=&x[k_1,k_2]+0.5x[k_1,k_2-1]+0.5x[k_1-1,k_2]+0.25y[k_1-1,k_2-1]\\&+0.5y[k_1-1,k_2-2]+0.5y[k_1-2,k_2-1]-0.25y[k_1-2,k_2-2]\end{aligned}$$

10.10 已知某二维系统的传输函数

$$H(z_1,z_2)=\frac{y(z_1,z_2)}{x(z_1,z_2)}=\frac{b_{1,1}z_1^{-1}z_2^{-1}}{1-a_{1,0}z_1^{-1}-a_{0,1}z_2^{-1}-a_{1,1}z_1^{-1}z_2^{-1}}$$

试求差分方程。

10.3 节

10.11 将脉冲响应 $h=[1,\ 2,\ 2]$ 与其自身相乘构造二维 FIR 滤波器。利用 MATLAB 画出在 65×65 矩阵空间上的滤波器的频率响应图。

10.12 利用 MATLAB 画出下列二维 FIR 滤波器的频率响应图：

(a) $H(z_1,\ z_2)=\dfrac{z_1^2z_2^2-z_1^2-z_2^2+1}{z_1^2z_2^2}$

(b) $H(z_1,\ z_2)=\dfrac{z_1^2z_2^2+z_1^2z_2+z_1^2+z_1z_2^2+z_1z_2+z_1+z_2+z_2^2+1}{z_1^2z_2^2}$

(c) $H(z_1,\ z_2)=1+2(z_1^{-1}+z_2^{-1})-2(z_1^{-2}+z_2^{-2})+4z_1^{-1}z_2^{-1}-4z_1^{-1}z_2^{-2}-4z_1^{-2}z_2^{-1}+4z_1^{-2}z_2^{-2}$

10.13 利用 MATLAB 画出下列二维 IIR 滤波器的频率响应图：

(a) $H(z_1,\ z_2)=4\dfrac{z_1z_2+0.5z_1+0.5z_2+1}{z_1z_2-2z_1-2z_2+4}$

(b) $H(z_1,\ z_2)=\dfrac{z_1^2+z_2^2+z_1z_2+z_1+z_2}{z_1^2z_2^2-0.2z_1^2-0.2z_2^2+0.04}$

10.14 下面的 MATLAB 程序运行后显示一个名为“im”的三维 sinc 函数图像。根据下列两个步骤计算 2D FFT：

步骤 1：对 im 的每一行进行 1D FFT 计算，生成新的图像 imRows。

步骤 2：对 imRows 的每一列进行 1D FFT 计算，生成新的图像 imFFT。

画出图像 imFFT，并与直接采用 MATLAB 函数 fft2 计算所得的二维 FFT 结果进行比较。

```
N = 128;
im = zeros(N, N);
[x1 x2] = freqspace(N,'meshgrid');
r = sqrt(x1.^2 + x2.^2);
im = sinc(2*pi*r);
```

10.15　画出下列函数的二维 FFT 图，所有函数均满足：

```
N = 256;f = 20;im = zeros(N, N);
```

(a) 横条纹

```
for c = 1:N
  im(c, :) = square(2*pi*f*c/N);
end
```

(b) 正弦波纹

```
im = zeros(N, N);
[x1 x2] = freqspace(N, 'meshgrid');
r = sqrt(x1.^2 + x2.^2);
im = sin(2*pi*r);
```

(c) chirp 函数

```
im = zeros(N, N);
[x1 x2] = freqspace(N, 'meshgrid');
r = sqrt(x1.^2 + x2.^2);
T = 1/f;
t = 0:T:N*T-T;
for c = 1:N
  im(:, c) = chirp(t, .2, N*T, 2);
end
```

(d) 锯齿波

```
for c = 1:N
  im(:, c) = sawtooth(2*pi*.5*f*c/N);
end
```

10.16　两个信号定义如下，利用 DFT 的性质计算两者的线性卷积，并利用 MATLAB 提供的 conv2 函数验证计算结果。

n_2 \ n_1	0	1	2	3
2	**0.5**	**1**	**0.5**	
1	**1**	**2**	**1**	
0	**0.5**	**1**	**0.5**	

$x[n_1, n_2]$

n_2 \ n_1	0	1	2	3
2				
1	**1**	**1**		
0	**2**	**2**		

$y[n_1, n_2]$

10.17　例 10-4 中低通滤波器的脉冲响应为

n_2 \ n_1	0	1	2	3	4
4	---	---	---	---	---
3	**0**	**0**	**0**	**0**	---
2	**0.7**	**0.98**	**0.49**	**0**	---
1	**1.4**	**1.96**	**0.98**	**0**	---
0	**1**	**1.4**	**0.7**	**0**	---

若输入信号如下，利用卷积计算输出信号。

n_2 \ n_1	0	1	2	3	4
4	---	---	---	---	---
3	**0**	**0**	**0**	**0**	---
2	**0**	**0**	**1**	**0**	---
1	**0**	**1**	**0**	**0**	---
0	**1**	**0**	**0**	**0**	---

10.18　下面的 MATLAB 程序运行后，得到一个三维 sinc 函数及其图形，将其作为一个二维滤波器的脉冲响应。

```
N = 255;
[t1 t2] = freqspace(N, 'meshgrid');
r = sqrt(t1.^2 + t2.^2);
Hw = sinc(2*pi*r);
Hw(r > 1) = 0;
figure(1);clf;
mesh(Hw);
```

如果滤波器的输入是由下面的程序产生的高斯型点阵，试用卷积计算滤波器的输出，并解释输出结果。

```
s = zeros(N, N);
rx = randn(5000,1);ry = randn(5000, 1);
rx = fix((N-1)*(rx - min(rx))/(max(rx)-min(rx))) + 1;
ry = fix((N-1)*(ry - min(ry))/(max(ry)-min(ry))) + 1;
for i = 1:length(rx)
  s(rx(i), ry(i)) = 1;
end
figure(2);clf;
imshow(s);
title('Gaussian Dots');
```

10.19　下面的 MATLAB 程序运行后显示一张由水平线和垂直线构成的网格图，对该图像进行 2D FFT 变换。在变换后的图像中，手动去掉垂直线对应的频率分量，再进行 IFFT 变换重建图像。验证去除的频率分量是否正确。

```
N = 256;
Im1 = zeros(N, N);
steps = 16;
for i = 1:N/steps
  start = i*steps - steps + 7;
  im1(:,start:start+steps/(N/steps)) = 1;
end
im2 = imrotate(imv,90,'nearest','crop');
im = im1 + im2;
```

10.20　若对图 AP10.20 中每幅图像进行 2D FFT，预测结果图像的形状，并利用 MATLAB 计算二维 FFT 图像，验证预测结果。

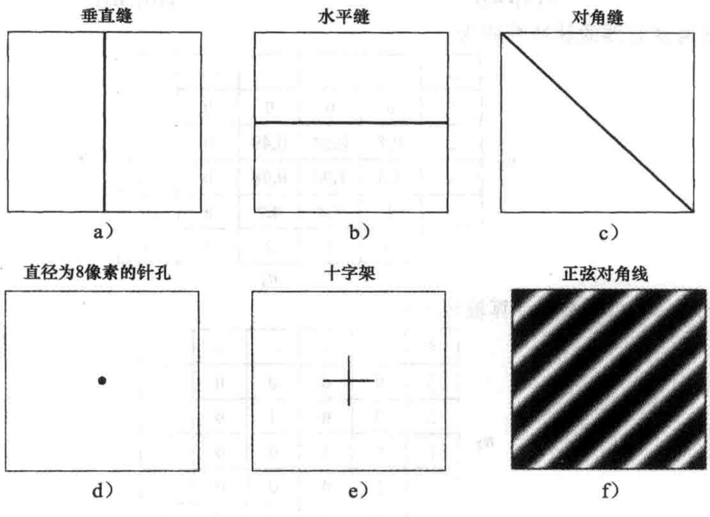


图 AP10.20　习题 10.20 图

10.21 利用 MATLAB 按如下步骤计算 $x[n]$的离散余弦变换

$$x[n]=\sin\left(\frac{\pi n}{2N}\right), N=8, n=0,1,2,\cdots,7$$

计算步骤：

1. 将 $x[n]$进行周期延拓得到 $x[n]=x[2N-n-1]$。
2. 对所得周期信号进行离散傅里叶变换得到 $X(K)$。
3. $X(K)$的每一项与 $p(k)e^{j(2\pi k/2)/(2N)}/2$ 相乘得到 $C(K)$。

调用 MATLAB 中的 dct 函数验证计算结果。

10.22 附录 D 中编制的 MATLAB 函数 fouriersq、fouriertri 和 fouriersaw 分别用于生成带限方波、三角波和锯齿波。调用这三个函数，生成的波形分别定义为 sq、tri 和 saw。每个信号都有 1024 个点，基频为 100Hz，抽样频率为 11025Hz，包含 28 项。

(a) 计算并画出每个信号的 DCT 图。

(b) 确定 DCT 的最少点数，使得其仍保留 95%的信号能量。

10.23 图 10.25 显示的二维汉明窗由一维窗函数旋转而得到，利用该方法构造一个长度为 65 的二维布莱克曼窗。

10.24 图 10.26 显示的二维汉明窗由两个一维窗相乘而得到。利用该方法构造一个长度为 33 的二维矩形窗。

10.25 构建一个二维理想低通滤波器，应用布莱克曼窗于理想滤波器，并显示结果。要求其归一化抽样频率为 2，截止频率为 0.18。

10.26 构建一个二维理想低通滤波器，应用汉宁窗于理想滤波器，并显示结果。要求其归一化抽样频率为 2，截止频率为 0.35。

10.27 构建一个二维理想高通滤波器，应用汉明窗于理想滤波器，并显示结果。要求其归一化抽样频率为 2，截止频率为 0.5。

10.28 添加必要的代码，验证下列加窗后的低通滤波器满足通带和阻带波动指标要求。

```
Rp = .05;Rs = .05;
fs = 2;
fpass = .2;fstop = .3;
fc = ((fpass + fstop)/2);
N = 128;
[f1 f2] = freqspace(N, 'meshgrid');
r = sqrt(f1.^2 + f2.^2);
Hd = zeros(N); %Ideal filter
Hd(r<fc) = 1;
Nw = 48; %Window size
zpad = zeros((N-Nw)/2, 1);
wHam = [zpad; hamming(Nw); zpad];
wHam2 = wHam*wHam';
h = fwind2(Hd, wHam2);
```

滤波器类型：低通加窗
抽样频率：2
通带和阻带波动：0.05
通带：0~0.2
阻带：0.3~1

10.29 试用布莱克曼窗设计满足下列指标的低通滤波器，要求滤波器阶数最低。

滤波器类型	低通加窗滤波器
抽样频率	2
通带波动	0.03
阻带波动	0.05
通带	0～0.33
阻带	0.5～1

10.30 试用汉宁窗设计满足下列指标的高通滤波器，要求滤波器阶数最低。

滤波器类型	高通加窗滤波器
抽样频率	2
通带波动	0.02
阻带波动	0.04
阻带	0～0.4
通带	0.5～1

10.31 试用汉明窗设计满足下列指标的带通滤波器，要求滤波器阶数最低。

滤波器类型	带通加窗滤波器
抽样频率	2
通带波动	0.05
阻带波动	0.03
阻带	0～0.3，0.62～1
通带	0.38～0.54

10.32 试用布莱克曼窗设计满足下列指标的带阻滤波器，要求滤波器阶数最低。

滤波器类型	带阻加窗滤波器
抽样频率	2
通带波动	0.03
阻带波动	0.01
阻带	0.4～0.6
通带	0～0.2，0.8～1

10.33 下面的MATLAB程序采用长度为28的矩形窗设计低通滤波器。试确定最大的通带波动，以滤波器增益百分比形式表示。

```
fs = 2;
fpass = .33;
fstop = .5;
fc = ((fpass + fstop)/2);
N = 128;
[f1 f2] = freqspace(N, 'meshgrid');
r = sqrt(f1.^2 + f2.^2);
Hd = zeros(N);
Hd(r<fc) = 1;
%
Nw = 28;
zpad = zeros((N-Nw)/2, 1);
wRect1 = [zpad; boxcar(Nw); zpad];
wRect2 = wRect1*wRect1';
h = fwind2(Hd, wRect2);
```

10.34 利用频率抽样法设计两个FIR滤波器。第一个滤波器的截止频率为0.5，并且没有过渡带。第二个滤波器有过渡带且通带截止频率为0.4，阻带截止频率为0.6。两个滤波器的抽样频率都为2，理想滤波器大小都为23×23点。画出两个滤波器响应的横截面图，显示两个滤波器响应的通带波动、阻带波动以及过渡带宽度，解释其差别。

10.35 设计两个满足如下指标的FIR滤波器。第一个滤波器采用频率抽样法实现，第二个滤波器采用汉明窗函数法实现。

滤波器类型	低通滤波器
抽样频率	2
通带波动	0.04
阻带波动	0.02
通带	0～0.43
阻带	0.65～1

比较两个滤波器的阶数。

10.36 已知某一维二阶低通FIR滤波器

$$H(z) = 0.3118\frac{z^2 + 1.46z + 1}{z^2}$$

(a) 求出脉冲响应，并表示成一个零相位系统。

(b) 将零相位系统的频率响应写成余弦函数的形式。

(c) 利用 McClellan 变换式(10.17)，将函数表示为二维 $\cos(\Omega_1)$和 $\cos(\Omega_2)$的幂次形式。

(d) 画出该滤波器的二维频率响应图。

10.37 试用帕克斯-麦克莱伦法设计满足以下指标的一维 FIR 滤波器。并将该滤波器变换到二维，显示变换后的结果。画出滤波器幅度响应的横截面放大图，验证设计的滤波器是否满足指标要求。

滤波器类型	低通滤波器
抽样频率	2
通带波动	0.02
阻带波动	0.02
阻带	0～0.28
通带	0.60～1

10.38 试用帕克斯-麦克莱伦法设计满足以下指标的一维 FIR 滤波器。并将该滤波器变换到二维，显示变换后的结果。画出滤波器幅度响应的横截面放大图，验证设计的滤波器是否满足指标要求。

滤波器类型	带通滤波器
抽样频率	2
通带波动	0.02
阻带波动	0.02
通带	0.35～0.45
阻带	0～0.20，0.60～1

10.39 例 10-17 中，MATLAB 程序生成的图像在频率为 0.25 处存在波纹，程序中抽样频率归一化为 $f_s/2=1$。试将该图像通过一个低通滤波器，并画出滤波结果。滤波器的通带边界频率为 0.18，阻带边界频率为 0.23。采用变换法和一维帕克斯-麦克莱伦低通滤波器设计该二维滤波器，通带和阻带波动为 0.02。

10.40 例 10-17 中，MATLAB 程序生成的图像在频率为 0.25 处存在波纹，程序中抽样频率归一化为 $f_s/2=1$。试将该图像通过一个高通滤波器，并画出滤波结果。滤波器的阻带边界频率为 0.22，通带边界频率为 0.28。采用变换法和一维帕克斯-麦克莱伦高通滤波器设计该二维滤波器，通带和阻带波动为 0.02。

10.41 下面的 MATLAB 代码产生一个近似正态分布的随机点阵图，如图 AP10.41 所示。

```
N = 256;
im3 = ones(N, N);
rx = randn(5000,2);
mx = max(max(rx));
mn = min(min(rx));
range = mx - mn;
rx = fix((N-1)*(rx - mn)/(mx-mn)) + 1;
for i = 1:length(rx)
  im3(rx(i,1), rx(i,2)) = 0;
end
imshow(im3);
title('Gaussian Dots');
```

a)



b)

图 AP10.41　正态分布的随机点

(a) 若将该图像通过窄带低通滤波器，解释图像将会发生什么变化。

(b) 若低通滤波器满足：$f_s=1$，$f_{pass}=0.05$，$f_{stop}=0.1$，通带和阻带波动都为 0.02。利用帕克斯-麦克莱伦滤波器和变换法设计二维滤波器，利用 MATLAB 编程画出滤波后的图像。

10.42　下面的 MATLAB 程序生成由 8 个圆构成的图像，如图 AP10.42 所示。

(a) 若该图像通过高通滤波器，解释图像将会发生什么变化。

(b) 若高通滤波器满足：$f_s=2$，$f_{pass}=0.40$，$f_{stop}=0.50$，通带和阻带波动都为 0.02。利用帕克斯-麦克莱伦滤波器和变换法设计二维滤波器，利用 MATLAB 编程画出滤波后的图像。

```
N = 256;
im1 = zeros(N, N);
rd = 16;
xc = [16 48 80 112 144 176 208 240];
yc = [32 32 96 96 160 160 224 224];
for i = 1:length(xc);
  for r = 1:N
    for c = 1:N
      if((c-xc(i))^2 +(r-yc(i))^2 < rd^2)
           im1(c, r) = 1;
      end
    end
  end
end
imshow(im1);
title('Circles');
```

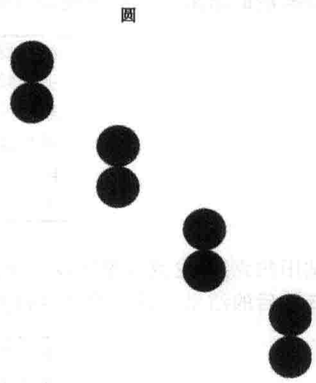
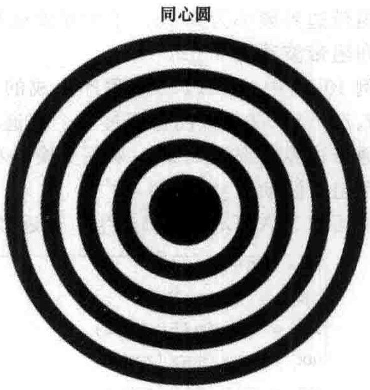


图 AP10.42　有序排列的圆

10.43　下面的 MATLAB 程序运行后生成黑白相间的同心圆环，如图 AP10.43 所示。若将该图像通过一个二维微分器，显示结果图像。二维微分器通过对理想二维微分器应用 $N=65$ 的汉明窗实现。

```
N = 256;
im2 = zeros(N,N);
ncircles = 16;
[f1 f2] = freqspace(N, 'meshgrid');
r = sqrt(f1.^2 + f2.^2);
i = 0;
for s = .1:.1:1
  im2(r > s) = mod(i, 2);
  i = i + 1;
end
figure(2);clf;
imshow(im2);
title('Concentric Circles');
```

a)



b)

图 AP10.43　同心圆

第11章 小波简介

小波从20世纪90年代进入数字信号处理(DSP)，其应用领域十分广泛，包括医学信号分析、数值分析，以及电磁信号、语音与图像信号和振动信号的分析等。小波分析渗透到诸多工程领域，且列入了工科本科生的课程。

本章简要介绍了小波—解释什么是小波、如何应用小波，以及相比于传统的傅里叶(Fourier)分析，小波分析有何优点，并给出了一些应用实例。

追溯小波的历史，小波与起源于19世纪早期的傅里叶级数和傅里叶变换有密切联系。傅里叶变换可以提供某给定信号的频率分析，其通过不同频率与幅度的正弦和余弦信号来表示待分析的信号。傅里叶变换只能给出信号中的频率分布，但难以给出信号中某指定频率出现的时刻。例如，某信号由一个频率为1kHz、持续时间为0.5s的正弦信号，并紧随另一个频率为2kHz、持续时间为0.5s的正弦信号而构成，傅里叶变换将给我们显示频率为1kHz和2kHz的频带，其中伴随一些由于该正弦曲线的截断而出现的其他一些更高频率的分量，但它无法告诉我们那个1kHz的信号在时域是位于前段还是后段。因此我们认为傅里叶变换比较适合于静态分析。目前已经有多种方法弥补傅里叶变换的上述不足，最著名的就是短时傅里叶变换(也称为加窗傅里叶变换)，小波分析提供了一种比传统短时傅里叶变换更佳的解决方法。

匈牙利数学家 Alfred Haar 于1909年在其博士论文中首先公开应用了第一个小波实例[1]。小波这个词在1909年时并不存在，今天我们将 Haar 当时引入的小波信号称为 Haar 小波。Haar 小波相对比较简单，便于我们引入小波的概念。小波这个术语首先由 Jean Morlet 在其出版物中采用[2]，他于20世纪80年代初期利用小波来分析地震信号。1987年 Ingrid Daubechies 在使得小波易于应用和理解方面做出了积极贡献[3]。

11.1 概述

根据第3章，描述傅里叶变换的方程表示为

$$F[x(t)] = X(\omega) = \int_{-\infty}^{\infty} x(t)\mathrm{e}^{-\mathrm{j}\omega t}\mathrm{d}t$$

利用欧拉(Euler)公式，可以用$[\cos(\omega t)-\mathrm{j}\sin(\omega t)]$替换 $\mathrm{e}^{-\mathrm{j}\omega t}$，得

$$F[x(t)] = X(\omega) = \int_{-\infty}^{\infty} x(t)\cos(\omega t)\mathrm{d}t - \mathrm{j}\int_{-\infty}^{\infty} x(t)\sin(\omega t)\mathrm{d}t$$

如果通过累加矩形面积来计算数值积分，上述方程变为

$$F[x(t)] = X(\omega) = T\sum_{n=-\infty}^{\infty} x(nT)\cos(\omega nT) - \mathrm{j}T\sum_{n=-\infty}^{\infty} x(nT)\sin(\omega nT) \tag{11.1}$$

余弦项求和可以用来测度信号 $x(nT)$ 与频率为 ω 的余弦函数之间的相关性(也就是说，如果某个频率 ω_0 的余弦函数与 $x(nT)$ 之乘积的累加和比较大，说明信号 $x(t)$ 中含有该余弦信号分量 $\cos(\omega_0 t)$ 较多)。同理，正弦项求和可以用来测度信号 $x(nT)$ 与频率为 ω 的正弦函数之间的相关性。符号 j 可以看作一个标识，用来分别跟踪这两个相关性。下面的例子表述了以相关视角展现傅里叶变换。

例 11-1　以相关性表述的傅里叶变换

产生一个信号，其前 50ms 是信号 $\cos(n\omega T-30°)$，后 50ms 为零。应用式(11.1)于该信号，计算结果可以测定该信号的幅度和相位。设定该信号中的频率为 100Hz，且抽样频率为 1000Hz。

解：利用 MATLAB 可以产生该信号

```
fs = 1000;T = 1/fs;
fsig = 100;
t = 0:T:.1-T;
x = cos(2*pi*fsig*t - pi/6);
x(length(t)/2:length(t)) = 0;
```

对于某特定的角频率 ω，可以根据式(11.1)计算得到下列值

```
sumCos = sum(cos(w*t).*x);
sumSin = sum(sin(w*t).*x);
mag = sqrt(sumCos^2 + sumSin^2);
phase = atan(-sumSin/sumcos)*180/pi;
```

这些方程可以计算出信号的幅度频谱图。如图 11.1 所示，其中，在 $\omega=2\pi(\text{fsig})$ 处相移为 $-30°$。

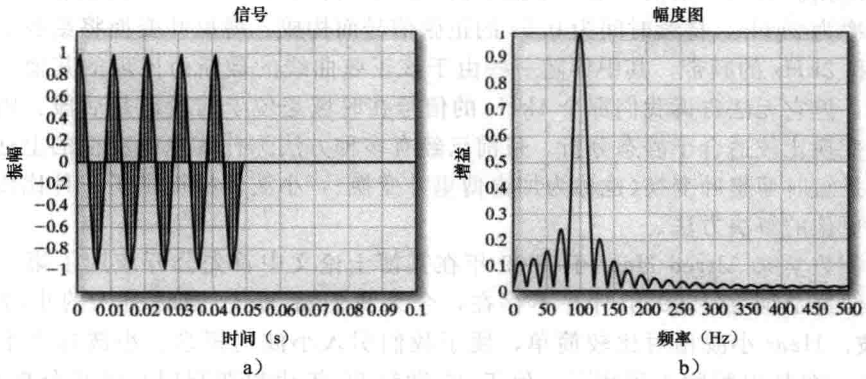


图 11.1　余弦函数和由式(11.1)得到的幅度图

关于例 11-1 有两点值得注意。首先，我们能够清晰地看到，信号从时域到频域的变换可以反映信号与正弦函数和余弦函数之间的相关性，且傅里叶变换中的正弦函数与余弦函数都是正交基函数。其次，在信号的幅度谱中，无法显现待分析信号的正弦曲线仅出现在其前半部分。也就是说，信号的幅度谱中没有任何时间信息。

小波变换与傅里叶变换的机理完全相同，傅里叶变换将信号表示为正弦函数和余弦函数，而小波变换将信号表示为小波函数。同理可以由小波变换反映信号与小波函数之间的相关性。两者的主要差别是，小波函数一般是时域有限长的信号(正弦和余弦函数是时间无限长信号)，因此，在小波变换域中可以同时获得时间信息和频率信息。

为了表述小波的特点，以图 11.2 所示的墨西哥帽小波为例进行表述。

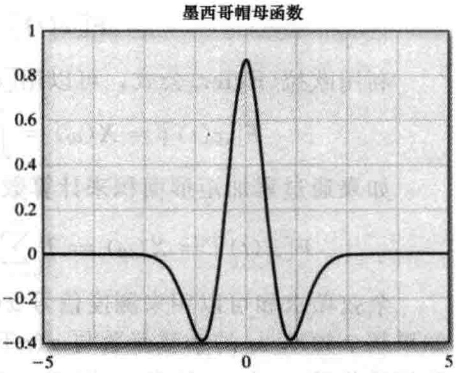


图 11.2　墨西哥帽小波母函数

当进行小波变换时，将小波母函数与信号相乘，然后求其和。由于小波信号是时域有限的信号，在乘积过程中，它与信号只有少部

分重叠。因此，需要平移小波信号至不同的时间段，再分别与信号进行乘积求和。小波信号每平移一段，就可以得到对应该时间段的小波信号与信号的相关性测度。图11.3显示了不同尺度(scale)下小波变换的结果，不同尺度的小波信号可通过对小波母函数进行展缩而得到。其中首先是信号在墨西哥帽小波于尺度1下的小波变换，其对应于尺度1下的墨西哥帽小波与信号乘积求和而得到的相关值，然后右移该小波信号，同理得到该时间段的相关数。将不同时间段对应的相关数以亮度级完整地描述出来，就形成了尺度1下的信号小波变换示意图。值得注意的是，小波函数的尺度对应于墨西哥草帽小波的频带。

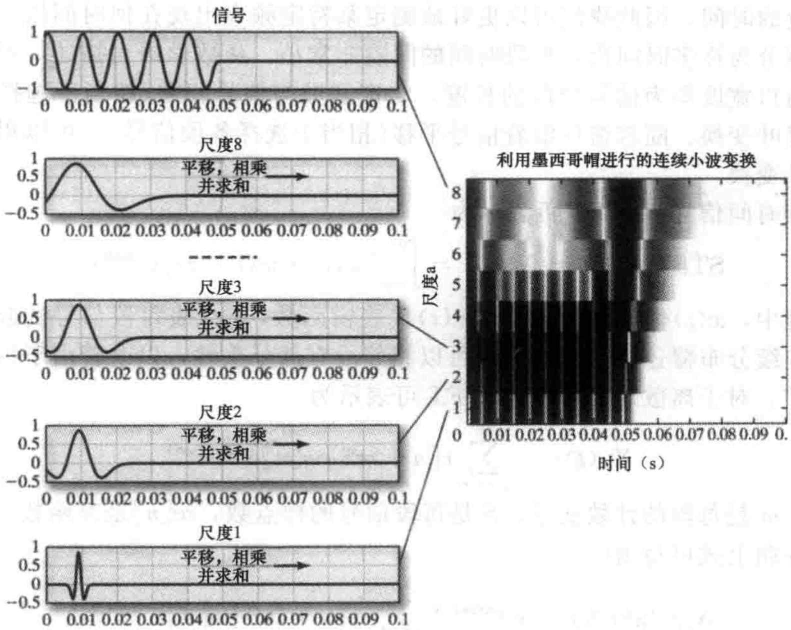


图11.3 顶部的信号分别与某个尺度的小波函数乘积求和，不同尺度下的小波信号对应不同的频带。小波函数从左向右平移，每次平移后小波函数与信号乘积求和得到相应值。右边显示了信号连续小波变换的示意结果

只需要将小波信号时域扩展一倍，就可以获得尺度2下信号的小波变换。由于小波信号在时域扩展一倍，其频谱在频域相应地压缩一倍。此例中，小波信号的尺度从1变化到8。从图11.3中清晰可见，最好的相关出现在尺度2附近，而且时间轴也较好地保留了信号的时间特性。该正弦波信号只持续到一半，信号停止造成信号改变的时间点，在不同尺度下的小波变换中都有呈现。

由于这是一个简化的小波变换的过程示意，因此仅利用了8个尺度。实际上，可以利用100以上的尺度以获得更高的频率分辨率。值得特别关注的是，图11.3中的纵轴标注的是“尺度”，用它代表频率。尺度与频率有关，但所有小波函数的频谱不是对应一个单频，而是对应一个频带。若要细分信号的频率分量，需要采用傅里叶变换。

图11.3描述了信号的连续小波变换(Continuous Wavelet Transform，CWT)。本章也将引入信号的离散小波变换(Discrete Wavelet Transform，DWT)。如此命名CWT是因为其小波变换的尺度轴可以是连续的。而实际进行小波变换的信号可能是离散的，因此连续小波变换需要转变为数字计算。DWT具有二分的离散尺度，其尺度按照2的幂次递变。我们将会看到，DWT通过对信号进行升抽样(内插)和降抽样(抽取)来体现小波函数的展缩。

我们通过简述短时傅里叶变换(Short-Term Fourier Transform，STFT)来介绍小波分析，并引入信号的时间一频率概念。从短时傅里叶变换引入小波变换，可以更好地衬托出小波的特性。

11.2 STFT

傅里叶变换假定信号是平稳的，即信号的频率不随时间而变化。对于非平稳信号，傅里叶变换只能给出信号的频率分布，但无法给出信号在何时发生了频率的改变。解决此问题的途径之一是将信号拆分为多段，分别对每段信号进行傅里叶变换。由于我们事先知道每段信号对应的时间，因此我们可以更好地测定某特定频率出现在何时间段。在理想情况下，将信号拆分为许多时间段，每段时间的间隔非常小。从数学语言描述，可以定义一个窗函数，其窗口宽度即为信号分段的长度。将窗函数与信号相乘(相当于选择某段信号)，然后进行傅里叶变换。随着窗口沿着信号平移(相当于选择各段信号)，可以得到各段信号对应的傅里叶变换。

对于连续时间信号，STFT 可表示为

$$\mathrm{STFT}(\tau,f)=X_s(\omega)=\int_{-\infty}^{\infty}[x(t)\cdot w(t-\tau)]\mathrm{e}^{-\mathrm{j}2\pi ft}\,\mathrm{d}t$$

在此方程中，$x(t)$代表输入信号，$w(t)$代表窗函数，τ代表每段信号的起点。窗函数可以无重叠连续分布覆盖整个信号，也可以相邻段有部分重叠，以改善由于拆分信号造成的“人为误差”。对于离散时间信号，STFT 可表示为

$$X_m(\Omega)=\sum_{n=-\infty}^{\infty}x[n+mS]w[n]\mathrm{e}^{-\mathrm{j}\Omega(n+mS)}$$

方程中，m 是每段的计数变量，S 是每段信号的样点数，$w[n]$是窗函数。通过适当运算后，可以证明上式可写为[4]

$$X_m(2\pi k/N)=\mathrm{e}^{-\mathrm{j}mS2\pi k/N}\sum_{n=-N/2}^{N/2}x[n+mS]w[n]\mathrm{e}^{-\mathrm{j}2\pi kn/N}$$

此方程可以看作加窗的时移输入信号的 DFT 再乘以 $\mathrm{e}^{-\mathrm{j}mS2\pi k/N}$。因此，可以利用 FFT 来计算 STFT。在 MATLAB 中，函数 spectrogram 可以实现此计算。

例 11-2 一个声音文件的短时傅里叶变换图形(Spectrogram)

利用 MATLAB 读入一个名为“doorbell. wav”的文件，产生该声音信号的一个短时傅里叶变换图形。文件中的门铃是类似“bing－bong”的传统双音调，信号的持续时间约 2.5s。

解： wavread 函数读取 WAV 文件，其将信号样点值、抽样频率和每个样点的位数返回给输出变量。

```
[y fs b]=wavread('doorbell.wav');
```

函数 spectrogram 计算信号的 STFT，并绘出其结果图。如果我们配合使用 mesh 函数，则可以对 spectrogram 函数的输出做出更多的操作。这里使用的 spectrogram 函数的输入变量依序如下：

y——输入信号

win——约定的汉明窗的宽度

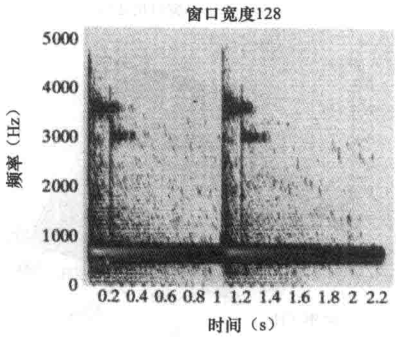
win/4——相邻窗口的重叠量

nfft——FFT 点数

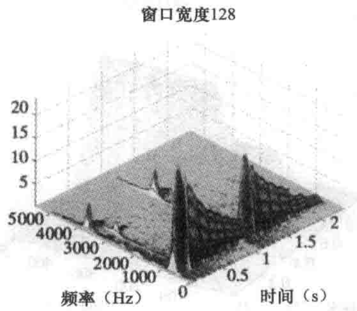
fs——抽样频率(Hz)

```
win = 128;
nfft = 128;
[s fr tr] = spectrogram(y, win, win/4, nfft, fs);
mesh(tr, fr, abs(s));
```

计算结果如图 11.4 所示。



a）数据的俯视图



b）数据的3D显示

图 11.4 门铃 WAV 文件的频谱图

由图 11.4 可见，每个音调由一个持续时间约 1s 且频率约 800Hz 的音符组成，并伴随一个持续时间约 0.1s 且频率约 3900Hz 的极短音调。◀

STFT 的主要问题是如何平衡频率分辨率与时间分辨率。为了得到更高的时间分辨率，我们需要将信号在时域尽可能分得越短越好。而为了得到更高的频率分辨率，最好不分段，因为待分析的信号持续时间越长，可以得到越高的频率分辨率。

考虑下面给定的信号

$$f(t) = \begin{cases} \sin(2\pi f_1 t), & 0 \leqslant t < 256 \\ \sin(2\pi f_2 t), & 256 \leqslant t < 512 \end{cases} \tag{11.2}$$

式中，$f_1 = 0.3\text{Hz}$，$f_2 = 0.7\text{Hz}$。

取抽样频率 $f_s = 2\text{Hz}$，即抽样周期 $T = 0.5\text{s}$，抽样后的信号如图 11.5 所示。

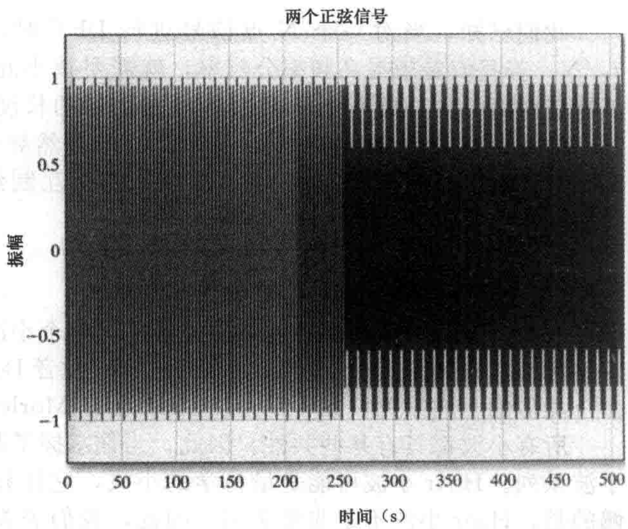


图 11.5 两个正弦信号。第一个正弦信号的频率为 0.3Hz，持续时间为 256s，第二个正弦信号紧随第一个正弦信号，其频率为 0.7Hz，持续时间为 256s

我们利用 MATLAB 分析此信号的 STFT，分段长度分别取 16 点、32 点、64 点和 256 点。由于抽样后的信号共有 1024 个样点，对应上述分段长度，可以分别得到 64 段、32 段、16 段和 4 段的信号。对所有分段情况，都采用汉明窗进行 STFT，而且相邻段之间采用 50% 的重叠。图 11.6 显示了该信号 STFT 的结果。

由图 11.6a 可见，我们获得了很好的时间分辨率，但频率分辨率相对较差。在时间轴上，信号低频部分在 256 个样点处锐截止。而在频率轴上，我们发现信号中的 0.3Hz 频率

是以 0～0.6Hz 的频带呈现。由图 11.6d 可见，效果恰好相反。由于信号只分为 4 段，因而频率轴方向得到了非常好的分辨率。信号中 0.3Hz 的低频由一个以 0.3Hz 为中心的窄带表示，而时间轴方向上的分辨率明显下降。时间轴上没有在相应位置(即第 256 个样点处)出现锐截止，而是呈现了一个在样点 180～320 之间的线性过渡。

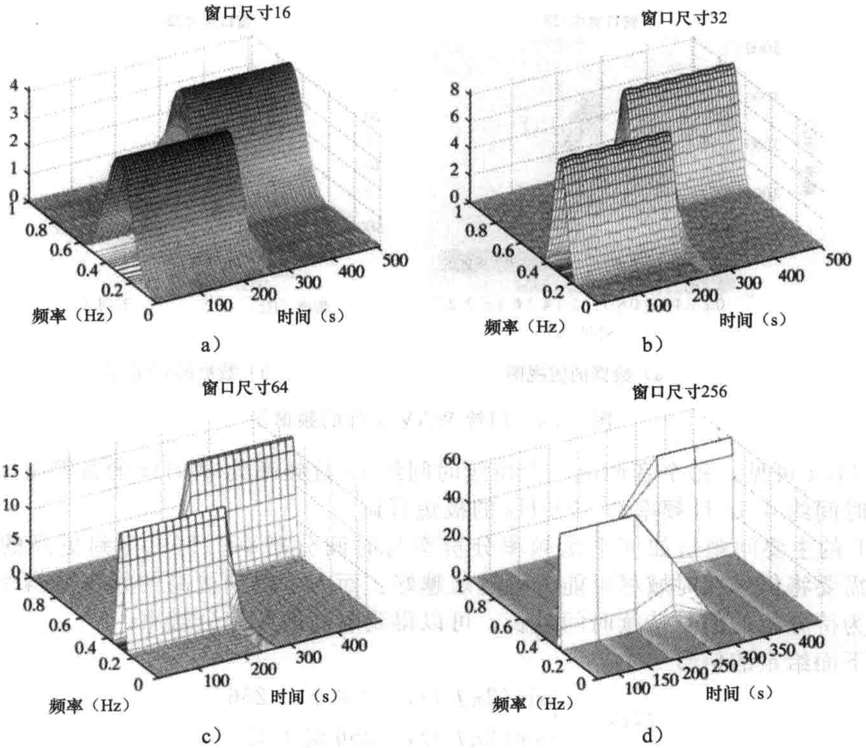


图 11.6　不同分段长度下式(11.2)定义信号的频谱图

我们已知，当对一个 N 点信号进行 DFT 时，我们可以在变换域得到频率间隔为 f_s/N。若我们希望提高频率分辨率，就需要减小此频率间隔。在实际应用中，我们通常不能任意降低抽样频率 f_s，而只能增加信号的长度 N。由此可见，在信号 STFT 中，若我们增加 N，即增加每段的信号长度，则必然导致时间分辨率的降低。因此，信号的 STFT 面临着时间分辨率和频率分辨率的相互制约。小波以不同的视角有效地化解此问题。

11.3　小波及其连续小波变换

小波一般以小波簇或小波系列来定义，每个小波系列都有其特点和适用对象。两个最常见的小波系列是 Haar 小波和 Daubechies(发音 Doab chee)小波。MATLAB 目前提供 10 多个类似的小波系列，如 Meyer、Gaussian、Morlet、Shannon 和 Mexican Hat 小波。

所有小波都具有某些共性。因此，当你掌握了某小波系列时，就可以很容易推及其他小波系列。Haar 小波可能是最易学的小波，它计算简单且拥有小波的所有基本特性。遗憾的是，Haar 小波不是非常实用。因此，我们需要了解其他小波来加深理解，并通过应用来感受小波的机理。

回想以前学过的傅里叶变换，其定义为

$$F(\omega) = \int_{-\infty}^{\infty} f(t)e^{-j\omega t}dt$$

傅里叶变换在时域将信号与复指数信号相乘，通过计算后得到另一个称为频域的信号。该变换利用了复指数信号所具有的正交特性，当信号中含有与某特定频率复指数信号相同的频率时，信号与复指数信号乘积的积分具有最大值，依此可以获得信号中频率分量的分布(即信号的频谱)。小波变换与傅里叶变换具有相同的机理，主要区别是，小波是持续时间有限的信号，而复指数信号是持续时间无限的信号。小波的持续时间有限带来两个结果：一是时间有限使得我们可以在小波变换中提取到信号的时间特性；二是时间有限的小波信号只能覆盖部分信号，因此需要平移小波信号以分段覆盖到全部信号。

每个小波都有其特定的波形，为了能够从小波与信号的乘积中提取到信号的频率分布信息，我们需要改变小波的波形。通过信号的尺度变化(即展缩)可改变小波的波形，以此来改变小波信号的频率分量。

11.3.1　展缩与平移

平移与展缩是两种简单的信号运算，可以应用到任意信号。考虑下列给定的信号

$$f(t) = \sin\left(\frac{t-b}{a}\right) \tag{11.3}$$

式中，尺度参数 a 实现信号在时域的展缩，而平移参数 b 实现信号在时域的平移。如图 11.7所示。

由图 11.7 清晰可见，当改变平移参数时(第 2 列)，信号的频率分布不变。而当改变尺度参数时(第 1 列)，信号的频率分布确实发生了改变。在此情况下，我们在时域通过时间变量 t 除以尺度参数以实现信号展缩，而信号时域的压缩对应信号频域的扩展，反之亦然。

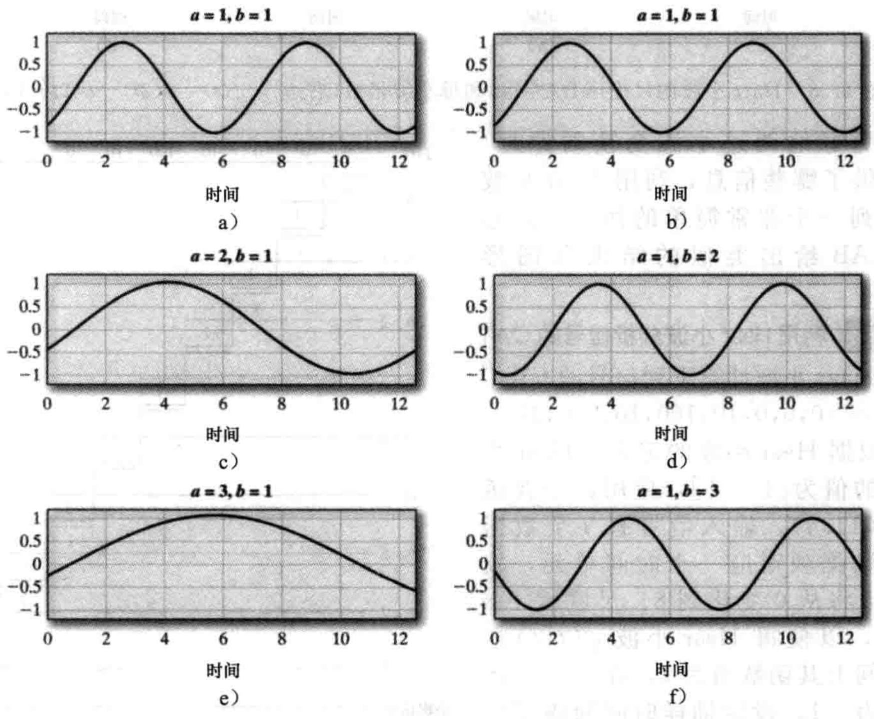


图 11.7　式(11.3)定义的函数，对应不同的尺度参数 a 和平移参数 b

CWT 以下列方程定义

$$W(a,b) = (1/\sqrt{a})\int_{-\infty}^{\infty} f(t)\psi^{*}\left(\frac{t-b}{a}\right)\mathrm{d}t \tag{11.4}$$

式中，* 是复数共轭，变量 a 是尺度参数，b 是平移参数。函数 ψ 是母小波函数。

小波信号由两个函数描述，分别称为尺度函数 $\varphi(t)$ 和母小波 $\psi(t)$。（在某些文献中，尺度函数也称为父小波。）

11.3.2 Haar 小波

Haar 小波对应的尺度函数与母小波可分别定义如下：

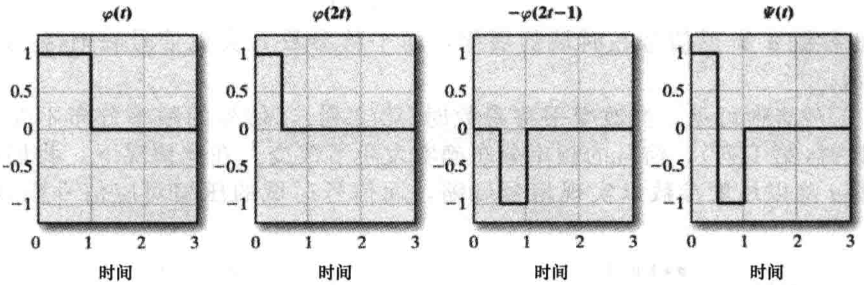
$$\varphi(t)=\begin{cases}1, & 0\leqslant t\leqslant 1\\ 0, & \text{其他}\end{cases}\quad \text{Haar 尺度函数}$$

$$\psi(t)=\begin{cases}1, & 0\leqslant t<1/2\\ -1, & 1/2\leqslant t<1\\ 0, & \text{其他}\end{cases}\quad \text{Haar 母小波}$$

对于此 Haar 小波，尺度函数与母小波之间存在下列关系：

$$\psi(t)=\varphi(2t)-\varphi(2t-1)$$

图 11.8 以图形方式显示了两者之间的关系。



a) b) c) d)

图 11.8 Harr 小波的尺度函数(左边)和母小波函数(右边)，$\psi(t)=\varphi(2t)-\varphi(2t-1)$

为了更好地阐述小波变换的原理，以及其提供了哪些信息，利用 Haar 小波来分析下列一个非常简单的例子，并通过 MATLAB 给出类似的结果和图形显示。

例 11-3 利用 Haar 小波分析信号的 CWT

利用 Haar 小波计算下列信号的 CWT $f(nT)=\{0,0,0,10,100,10,0,0,0\}$

解： 根据 Haar 小波的定义，Haar 小波滤波器的值为[1 −1]，应用此小波函数 ψ 至式(11.4)。输入信号有 9 个数据项，每个数据项对应一个时间单元，故平移因子 b 将从 0 变化到 8。尺度因子 a 从 2 开始，以使得 Haar 小波 $\psi(t/2)$ 在[0，1]区间上其函数值为 1，在[1，2]区间上其值为 −1。设定抽样时间间隔 $T=1$，利用式(11.4)分别对每个平移参数求其对应的数值积分。计算过程如图 11.9 所示。

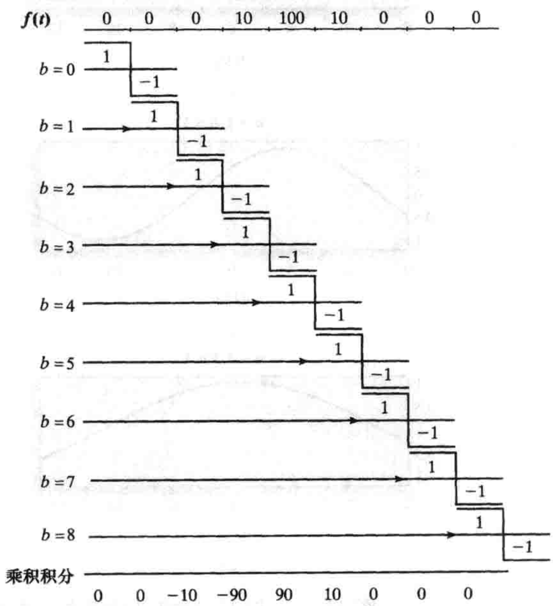


图 11.9 Haar 小波分别向右平移 b 个样点，然后与信号 $f(t)$ 相乘，求其积分，在底部显示其积分值。如 $b=3$ 时，可以得到 $1\times10+(-1)\times100=-90$

在图 11.9 中，尺度因子 $a=2$，根据

式(11.4)我们必须将计算结果乘以$1/\sqrt{2}$。最终得到的信号CWT的结果为

$$W(2,b)=\{0,0,-7.0711,-63.6396,63.6396,7.0711,0,0,0\} \tag{11.5}$$

上述实现了信号在尺度因子$a=2$下的CWT计算。同理，可计算尺度因子$a=3$时的CWT，只不过需要通过尺度因子$a=3$扩展Haar小波。用$t/3$替代Haar母小波信号定义式中的t，可得

$$\psi(t/3)=\varphi(2t/3)-\varphi(2t/3-1)$$

根据此尺度扩展后的小波信号$\psi(t/3)$定义，其在区间[0，1.5]上的函数值为1，在区间[1.5，3]上的函数值为−1。设定在区间[0，1]上的函数值为+1，在区间[2，3]上的函数值为−1，而在区间[1，2]上的函数值取该区间上的信号平均值0。扩展后的Haar小波信号的值为[1 0 −1]。对应该尺度因子的CWT计算如图11.10所示，最后所得结果需要除以$\sqrt{3}$，其结果为

$$W(3,b)=\{0,0,-5.7735,-57.7350,0,57.7350,5.7735,0,0,0\}$$

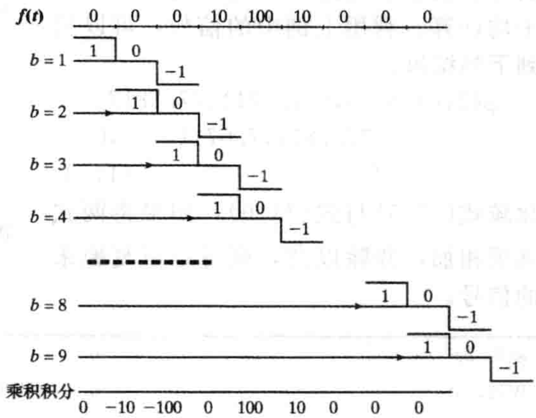


图11.10 $W(3,b)$的计算，最后的乘积积分值需除以$\sqrt{3}$得到信号的CWT

该计算过程可推至更大的尺度因子。MATLAB提供一个函数cwt来计算信号的CWT。下列语句即可求解此问题。

```
f = [0 0 0 10 100 10 0 0 0];
c = cwt(f, 1:10, 'haar', 'plot');
```

向量c中保留了计算结果，其行表示尺度，列表示时间。表11.1和图11.10显示了计算结果。图11.11绘出了表11.1的数据。

表11.1 $f[n]=\{0,0,0,10,100,10,0,0,0\}$的Haar小波变换系数。该结果由MATLAB的cwt函数产生。注意，MATLAB函数在变换前，在数据的两端进行了补零

尺度值 \ 时间值	1	2	3	4	5	6	7	8	9
1	0	0	0	0	0	0	0	0	0
2	0	0	0	-7.07	-63.64	63.64	7.07	0	0
3	0	0	-5.78	-57.83	0.15	57.70	5.77	0	0
4	0	0	-5.0	-55.0	-50	50	55.0	5.0	0
5	0	-4.48	-49.24	-49.22	0.22	49.24	49.02	4.45	0
6	0	-4.09	-45.0	-49.06	-40.67	40.93	49.06	44.76	4.07
7	-3.80	-41.75	-45.29	-41.47	0.26	41.52	45.27	41.49	3.77
8	-3.54	-38.89	-42.43	-42.43	-35.36	35.36	42.43	42.43	38.89
9	-36.74	-408	-408	-36.71	0.29	36.77	40.08	40.05	36.45
10	-34.96	-37.86	-38.11	-37.83	-31.47	31.81	37.86	38.11	37.83

读者可能注意到例11-3中的信号并非连续信号，但我们却对其进行了连续小波变换CWT。连续小波变换并非要求信号必须为连续信号。实际上，这里"连续"这个词是指尺

度因子可以取任意值。而相应的离散小波变换DWT中，其尺度因子只取2的幂次，因而极大地降低了小波变换的计算量，此称为二分(dyadic)尺度。

原先的例题利用Haar小波滤波器。我们也可以利用Haar尺度函数滤波器。该尺度函数具有系数$\varphi(t)=[1\ 1]$，因而其可以按照距离为2实现对信号的滑动平均计算。利用上例中的信号，可以得到下列结果：

$$\varphi(2,b)=\{0,0,7.0711,77.7817,77.7817,7.0711,0,0,0\} \tag{11.6}$$

比较式(11.5)与式(11.6)，如果将两式逐项相加，并除以$\sqrt{2}$，则可以恢复原来的信号。

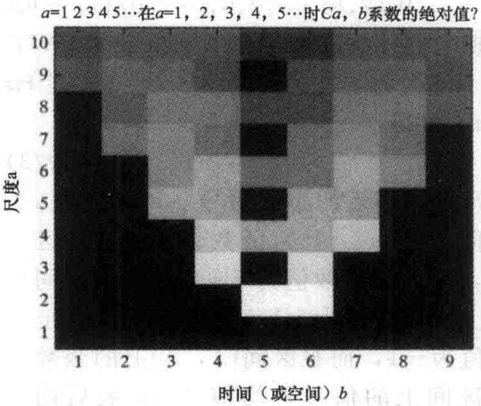


图 11.11　表11.1数据的图像显示。较高的频率对应较低的尺度值。较亮的灰度对应较大的系数值

$\varphi(2,b)$	0	0	7.071	77.782	77.782	7.071	0	0
$W(2,b)$	0	0	−7.071	−63.640	63.640	7.071	0	0
$(\varphi(2,b)+W(2,b))/\sqrt{2}$	0	0	0	10	100	10	0	0

尽管这是一个简单的例子，但确阐述了一个基本原理。更一般地，可以利用下列方程恢复信号$f(t)$。

$$f(t)=\sum_{k=-\infty}^{\infty}c_k\varphi(t-k)+\sum_{k=-\infty}^{\infty}\sum_{j=0}^{\infty}d_{j,k}\psi(2^jt-k) \tag{11.7}$$

式中，c_k表示尺度函数系数，$d_{j,k}$表示小波系数。

11.3.3 Doubechies 小波

Haar小波是Doubechies小波系列中最简单的一个成员，它也是唯一可以闭合形式描述的小波。在MATLAB中，Haar小波也称为db1。Doubechies小波系列的下一个成员在MATLAB中称为db2[⊖]，其由下列4点序列定义为{−0.1294,−0.2241，08365，−0.4830}。这4点构成小波滤波器，如图11.12所示。对应的尺度滤波器为{0.4830，0.8365，0.2241，−0.1294}。我们将在后面讨论DWT时解释这4个点的由来。

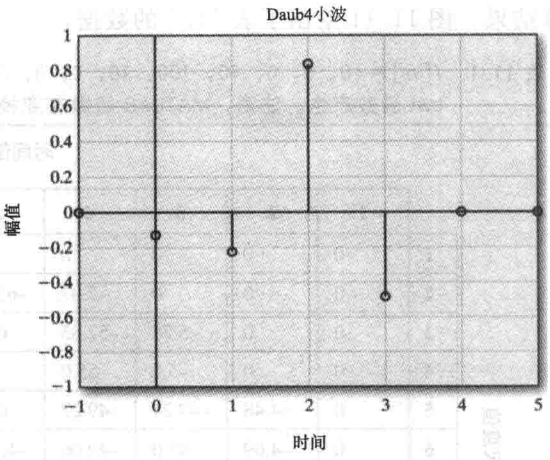


图 11.12　Daub4小波函数由此4点定义{−0.1294，−0.2241，08365，−0.4830}

由于Daub4小波无法以闭合形式定义，因此需通过其他方式产生对应的尺度函数。有

⊖ 在MATLAB中，其根据小波具有的消失矩(Vanishing Moment)数目来分类Daubechies小波。在文献中，更多地根据小波滤波器的点数来分类小波。例如，在MATLAB中的db2可能在别处以Daub4出现。消失矩数目等于对应的尺度滤波器在$f_s/2$处的零点数目。一个具有N个消失矩的滤波器与N阶多项式具有很高的相关性，其常用来估计噪声。

多个算法产生尺度函数，包括某些合理近似的算法。Daubechies 给出了一个函数以计算尺度系数[5]，M. J. Shensa 的论文[6]证明了 Daubechies 系数与 Lagrange à trous 滤波器之间的关系。Lagrange à trous 滤波器通过多项式 P 来定义，其系数为

$$\{a(N),0,a(n-1),0,\cdots,0,a(1),1,a(1),0,a(2),0,\cdots,0,a(N-1),0,a(N)$$

式中

$$a(k)=\frac{\prod\limits_{\substack{i=-N+1\\ i\neq k}}^{N}(1/2-i)}{\prod\limits_{\substack{i=-N+1\\ i\neq k}}^{N}(k-i)}$$

在求解得到多项式 P 的系数后，可以根据关系式 $P=\tilde{h}*h$ 得到 Daubechies 小波系数，其中，h 表示 Daubechies 小波系数，$\tilde{h}$ 表示 h 的时域翻转，$*$ 表示卷积。MATLAB 在函数 dbaux 中采用了该算法。此外，Fugal[7] 采用了一种内插滤波器的近似方法。首先给出一个 4 点滤波器，内插后再通过尺度滤波器滤波。重复此过程，得到一个近似的多点 Daubechies 小波滤波器。

MATLAB 提供了函数 wavefun，其可给出相应的尺度滤波器和小波滤波器。

例 11-4　计算 Daubechies 尺度和小波滤波器

利用 MATLAB 求解 db2、db4、db8 和 db16 的尺度滤波器和小波滤波器，并绘出其波形。

解：

```
wnum = [2 4 8 16];
figure(1);clf;
for i = 1:length(wnum)
    n = wnum(i);
    s = ['db' num2str(wnum(i))];
    [phi psi pts] = wavefun(s, 8);
    subplot(2, 2, i);
    plot(pts, psi);
    axis([0 max(pts) -1.5 2]);
    xlabel('time');
    ylabel('amplitude');
    title([s 'wavelet']);
end
```

图 11.13 显示了计算结果。

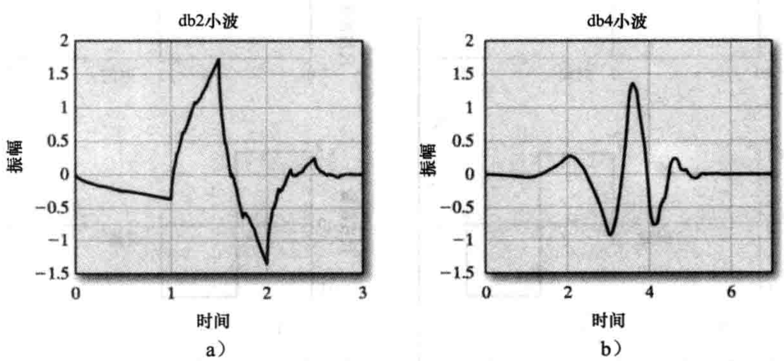


图 11.13　Daubechies 小波

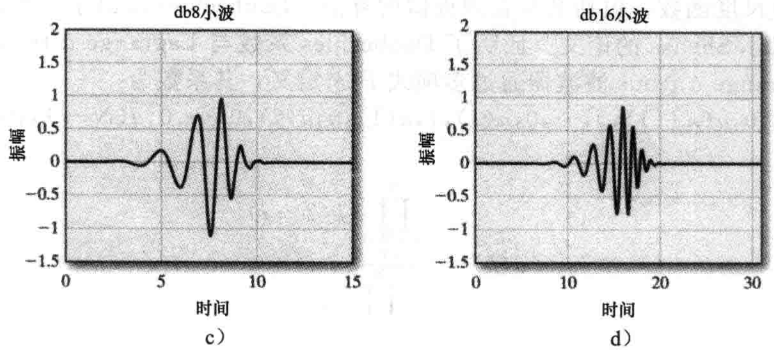


图 11.13 （续）

11.3.4　其他小波系列

目前大量的小波已经定义，MATLAB 为这里讨论的小波都提供了相应函数。小波按照其阶数或长度；其是否正交，双正交，或两者都不是；其对称性以及它们具有的消失矩数目等来分类。下面根据上述这些分类属性分别讨论其他小波系列。

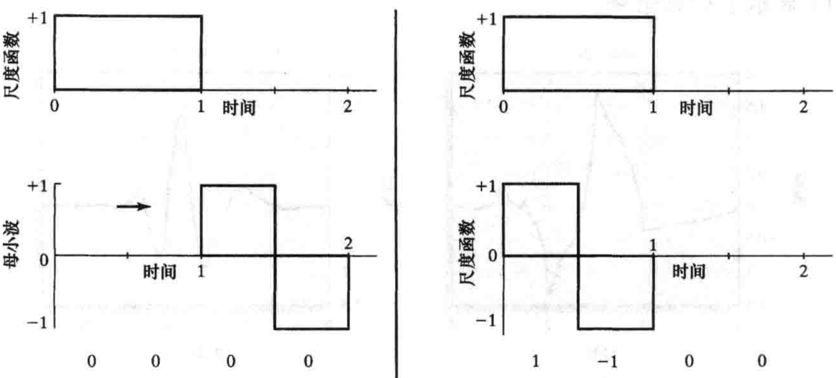
长度(length)——此一般是小波滤波器的点数。在多数文献中，Daubechies 小波按其长度分类。例如，Daub4 是指 4 点 Daubechies 小波。在其他场合(特别是在 MATLAB)中，Daubechies 小波按其具有的消失矩数目进行定义。Daub4 具有两个消失矩，因而在 MATLAB 中列入 db2 类。

紧支(Compact Support)——该术语“Compact Support”是指小波具有有限的长度。Haar 小波和 Daubechies 小波都具有紧支特性，而莫斯哥帽小波不是紧支小波。如果某小波函数在时域上比较窄，其将具有较好的时间分辨率，但具有较差的频率分辨率，反之亦然。

正交性(Orthogonality)——简言之，对于小波函数，正交是指当以小波函数乘以时间信号，且平移小波信号重复这样的乘积过程时，两者的乘积不包含重叠的信息。并非所有的小波都具有正交性，非正交的小波也有用途。如果尺度函数的系数设为 h_k，假如满足下列方程，则我们可以说对应的小波是正交的。

$$\sum_k h_k h_{k+2n} = 0, \quad n \neq k$$

Haar 小波提供了一个显而易见的正交性例子。由图 11.14 可见，只要母小波平移一



a）母小波平移一个整数，没有重叠　　b）母小波与尺度函数出现重叠，但内积为零

图　11.14

个整数，则内积为零。为了从数学上定义正交性，定义两个向量的内积(inner product)

$$内积 = \langle (x_1, x_2, \cdots, x_N), (y_1, y_2, \cdots, y_N) \rangle = x_1 y_1 + x_2 y_2 + \cdots + x_N y_N$$

如果两个向量的内积为零，则此两个向量正交。

小波的正交性具有重要意义，因为这样可以使得时间和频率两个变量具有独立性。所有 Daubechies 小波都具有正交特性。在图像压缩和图像编码等应用中，正交小波是较为理想的小波。

规范正交(Orthonomal)——当向量没有平移时，如果两个向量的内积为常数，而当向量平移一个整数时，其内积为零，则称它们为规范正交的。例如，Daub4 小波的尺度函数是{−0.0915，0.1585，0.5915，0.3415}。利用 MATLAB 计算其内积为

```
L = [-0.0915 0.1585 0.5915 0.3415];
disp(sum(L.*L));
```

得到答案为 0.5。假如平移向量 $\boldsymbol{L}$ 一个整数量，再计算其内积

```
LShift = [0 0 L];
LZero = [L 0 0];
disp(sum(LShift.*LZero);
```

得到的结果为零。我们说向量 $\boldsymbol{L}$ 对其自身是规范正交的。

双正交性(Biorthogonality)——对于一个正交的尺度函数，我们常将其表示为两个其他函数的卷积，对于母小波函数也可同样表示。此导致一个小波函数将对应两个尺度函数 φ 和 $\widetilde{\varphi}$，以及两个母小波函数 ψ 和 $\widetilde{\psi}$，它们可能正交也可能不正交。这样的小波称为双正交小波。具有对称特性的双正交小波在图像处理等方面具有良好的效果。

对称性(Symmetry)——对称性是指小波函数关于某个中心点是否对称(或可能的反对称)。例如，Meyer 和 Mexican Hat 小波是对称的。Haar 小波是唯一具有对称性、紧支性、正交性的小波。习题 11.26 将给出一个简要的证明。

消失矩(Vanishing moment)——消失矩数目为 V 的小波可以表示阶数为 $V-1$ 的多项式。对应的尺度函数滤波器将在 $f_s/2$ 处具有 V 个零点。信号能够以多项式表示很有意义。例 11-5 解释了消失矩的应用。

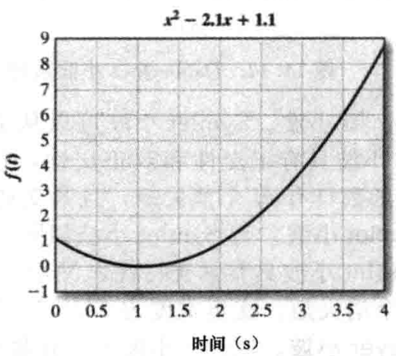
例 11-5 计算一个多项式函数的连续小波变换 CWT

考虑一个给定的多项式 $f(t)=t^2-2.1t+1.1$。对该多项式按抽样频率 100Hz 抽样，产生一个持续时间为 4s 的信号。利用 MATLAB 提供的 db1、db2 和 db3 分别计算该信号的 CWT。分析比较给出的小波变换的图示结果。

解：计算和绘制多项式和 CWT 的 MATLAB 代码如下，其结果如图 11.15 和图 11.16 所示。

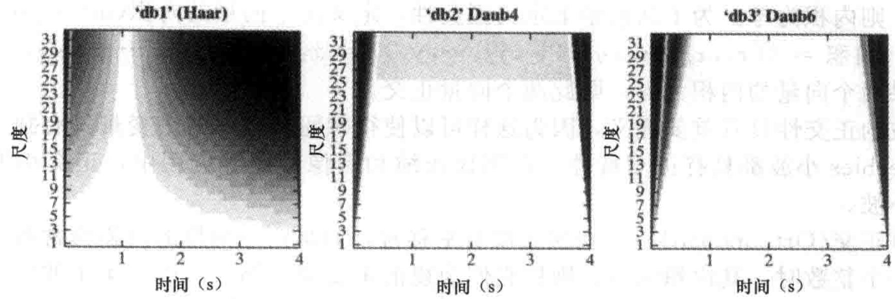
```
fs = 100;T = 1/fs;
t = 0:T:4;
f = @(x)(x.*x - 2.1*x + 1.1);
figure(1);clf;
plot(t, f(t));
s = 1:32;
figure(2);clf
subplot(1, 3, 1);
CWTcoeffs = cwt(f(t),s,'db1', 'plot');
subplot(1, 3, 2);
CWTcoeffs = cwt(f(t),s,'db2', 'plot');
subplot(1, 3, 3);
CWTcoeffs = cwt(f(t),s,'db3', 'plot');
```

a)



b)

图 11.15 显示 $x^2-2.1x+1.1$ 和三个小波对应的 CWT 的 MATLAB 代码



a）利用具有一个消失矩的db1(Haar)小波　　b）利用具有两个消失矩的Daub4小波　　c）利用具有三个消失矩的Daub6小波

图 11.16　多项式的连续小波变换(采用反向显示，越暗代表 CWT 值越大)

注意，只有一个消失矩的 Haar 小波未能匹配该多项式，因而我们看到许多非零的系数值。Daub4 小波有两个消失矩，其可以匹配一阶多项式。在此例中，匹配比较好但没有精确匹配。Daub6 小波有三个过零点，小波可以精确地匹配一阶和二阶多项式，因此此例中除了某些边缘效应外，系数几乎全部为零。◀

多分辨率(Multiresolution)——函数的多分辨率是指函数空间可以分解为子空间。对于 DWT，当时间轴按照 2 的幂次展缩时，意味着频率轴按照倍频显示。

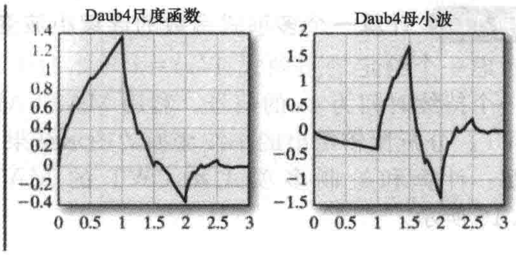
正则度(Regularity)——小波函数可以被连续微分的次数的近似测度。正则度与平滑度(Smoothness)常交替使用，小波的正则度越高，其越光滑。

下面简要介绍几种常见的小波。

Daubechies 小波。Daubechies 小波既是正交的也是双正交的，而且它们都具有紧支特性。*N* 阶 Daubechies 小波的长度为 2*N*，其紧支宽度为 2*N*−1。它们都不具有对称性(db1 除外)，而且其尺度函数具有 *N* 个消失矩。图 11.17 显示了 Daubechies 小波。

```
%Daubechies
[phi psi x] = wavefun('db2', 8);
figure(8);clf;
subplot(1, 2, 1);
plot(x, phi);
title('Daub4 Scaling Function');
subplot(1, 2, 2);
plot(x, psi);
title('Daub4 Mother Wavelet');
```

a）



b）　　c）

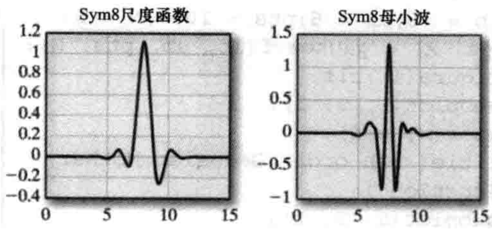
图 11.17　Daubechies 小波系列。此图显示了 Daub4 尺度函数(b)和母小波(c)

Symlet 小波。Symlet 小波可以从 Daubechies 小波推导而来，其具有近似的对称性。Symlet 小波具有正交性和双正交性，且具有紧支特性。*N* 阶 Symlet 小波的长度为 2*N*，其尺度函数具有 *N* 个消失矩，且紧支宽度为 2*N*−1。该小波如图 11.18 所示。

Coiflet 小波。与 Symlet 小波相同，Coiflet 小波具有正交性和双正交性，几乎是对称的。Coiflet 小波具有紧支特性。*N* 阶 Coiflet 小波具有的滤波器长度为 6*N*，其尺度函数具有 2*N* 个消失矩。支集宽度为 6*N*−1。该小波如图 11.19 所示。

Meyer 小波。Meyer 小波不具有紧支特性，但具有对称特性。Meyer 小波是正交的和双正交的。Meyer 小波的支集宽度为无限长，但其有效的支集宽度是−8～+8。该小波如图 11.20 所示。

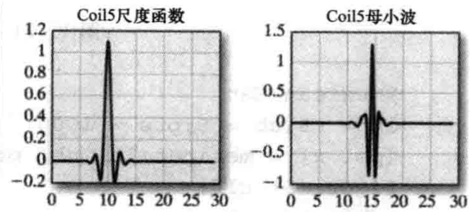
```
%symlet
[phi psi x] = wavefun('sym8', 8);
figure(1);clf;
subplot(1, 2, 1);
plot(x, phi);
title('Sym8 Scaling Function');
subplot(1, 2, 2);
plot(x, psi);
title('Sym8 Mother Wavelet');
```



a)　　b)　　c)

图 11.18　Symlet 小波。此图显示了 sym8 对应的尺度函数和母小波

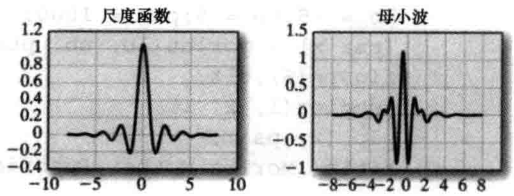
```
%Coiflet
[phi psi x] = wavefun('coif5', 8);
figure(1);clf;
subplot(1, 2, 1);
plot(x, phi);
title('Coif5 Scaling Function');
subplot(1, 2, 2);
plot(x, psi);
title('Coif5 Mother Wavelet');
```



a)　　b)　　c)

图 11.19　Coiflet 小波。此图显示了 coif5 对应的尺度函数和母小波

```
%Meyer
N = 8; %must be a power of 2
[phi psi x] = wavefun('meyr', N);
figure(3);clf;
subplot(1, 2, 1);
plot(x, phi);
title('Meyer Scaling Function');
subplot(1, 2, 2);
plot(x, psi);
title('Meyer Mother Wavelet');
```



a)　　b)　　c)

图 11.20　Meyer 小波

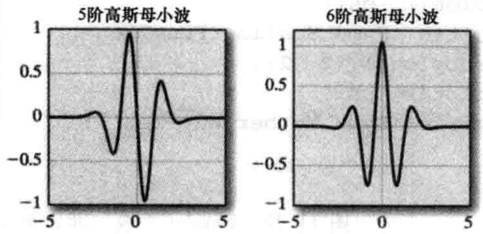
高斯小波。高斯小波既不是正交的也不是双正交的，且不具有紧支特性。它们的偶数阶是对称的(偶对称)，奇数阶是反对称的(奇对称)。高斯小波是基于 Gaussian 密度函数的一阶微分。该小波的紧支宽度为无限，但其有效的紧支宽度是－5～＋5。该小波如图 11.21所示。

墨西哥帽小波。墨西哥帽小波是基于高斯密度函数的二阶微分，其既不具有正交特性也不具有双正交特性。该小波是对称的，但不具有紧支特性。其有效的支集宽度是－5～＋5。该小波如图 11.22 所示。

Morlet 小波。Morlet 小波具有对称特性，其不具有紧支特性，但其有效支集宽度为－4～4。它们既不是正交的也不是双正交的。该小波如图 11.23 所示。

```
%Gaussian
lb = -5;ub = 5;pts = 1000;N = 5;
[psi x] = gauswavf(lb, ub, pts, N);
figure(4);clf;
subplot(1, 2, 1);
plot(x, psi);
title('5th order Gaussian Mother
Function');
subplot(1, 2, 2);
lb = -5;up = 5;pts = 1000;N = 6;
[psi x] = gauswavf(lb, ub, pts, N);
plot(x, psi);
title('6th order Gaussian Mother
Wavelet');
```

a)

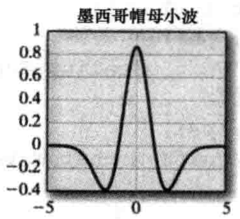


b)　c)

图 11.21　高斯小波

```
%Mexican Hat
lb = -5;ub = 5;pts = 1000;
[psi x] = mexihat(lb, ub, pts);
figure(5);clf;
subplot(1, 2, 1);
plot(x, psi);
title('Mexican Hat Mother Function');
```

a)

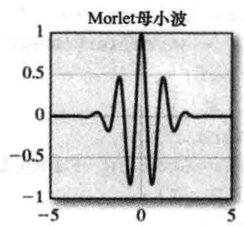


b)

图 11.22　墨西哥帽小波

```
%Morlet
lb = -5;ub = 5;pts = 1000;
[psi x] = morlet(lb, ub, pts);
figure(6);clf;
subplot(1, 2, 1);
plot(x, psi);
title('Morlet Mother Function');
```

a)



b)

图 11.23　Morlet 小波

11.4　小波变换数据的解释

频域。CWT 给出的系数是一个时间和尺度的函数，其尺度与频率有关。小波信号的频谱分布在一个频率范围(频带)，不同于作为傅里叶变换基信号的正弦和余弦信号，其频谱是单频(频点)。当将小波与某信号相乘时，我们得到的是信号与小波对应的频带相关的结果。因此，我们无法将某个频率与某个尺度相对应，只是通过计算某尺度下小波信号的 DFT，求其绝对值中的最大值，利用该频带中的幅度最大值近似代表该小波的频率点，该频率称为此小波信号的中心频率(center frequency)。

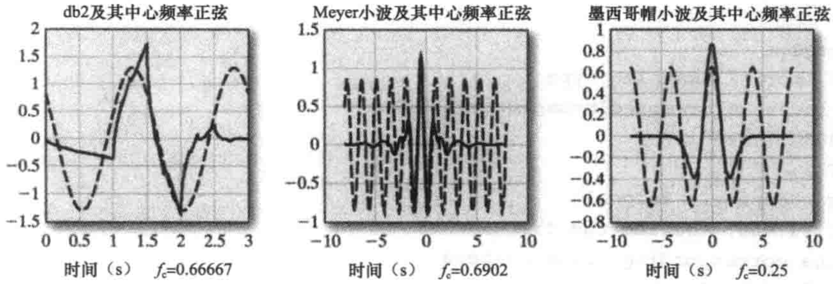
例 11-6　小波中心频率

在同一幅图中分别画出 Daub4、Meyer 和 Mexican Hat 母小波信号及其对应的中心频率。

解：MATLAB 提供了函数 centfrq，其可以计算指定小波的中心频率。该函数提供了一个选项以画出小波信号和以中心频率为频率的正弦信号。以下程序说明如何计算 db2 小

波信号的中心频率。图 11.24 给出了三种小波的计算结果。

```
figure(1);clf;
f = centfrq('db2', 8, 'plot');
haxis = gca; %Get Current Axis handle
axisTITL = get(haxis,'title');
axisXLAB = get(haxis,'xlabel');
set(axisTITL,'String','db2 and centered sinusoid');
s = (['time in seconds' 'fc = 'num2str(f)]);
set(axisXLAB,'String',s);
```

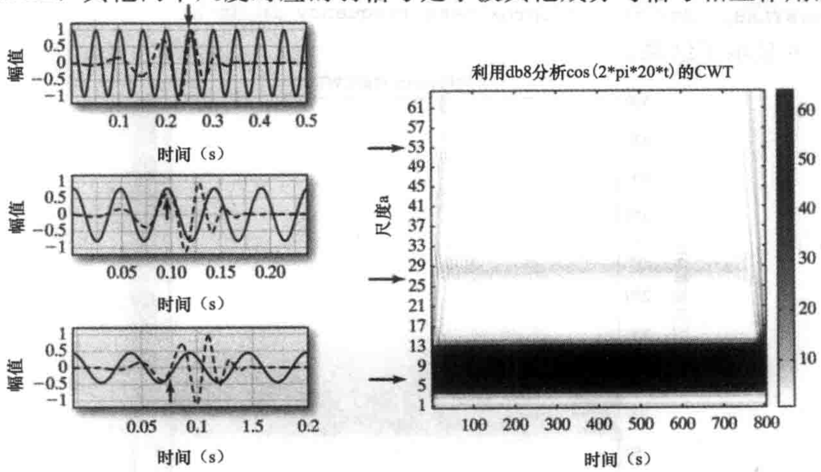


a）Daub4小波　　b）Meyer小波　　c）墨西哥帽小波

图 11.24　MATLAB 函数 centfrq 绘出的图形。虚线表示以中心频率为频率的正弦信号，实线表示母小波函数

显而易见，Meyer 小波和墨西哥帽小波在检测和分离某特定正弦信号方面优于 Daub4 小波。◀

值得注意的是，尽管频率与尺度有密切关系，但频率并不等于尺度。通过图 11.25 进行图形解释，对于一个周期为 0.05s(频率为 20Hz)的正弦信号，应用 Daub16 小波对其进行连续小波分析，发现其与三个尺度相匹配。正弦信号的表达式为 $f(t)=\cos(2\pi 20t)$，其对应的 CWT 如图 11.25b 所示。图中可见在尺度 7 处存在一个主频率，在尺度 28 处存在微弱频率，在尺度 53 处存在一个极弱频率。而输入信号只是一个单频信号。尺度 7 对应于信号频率 20Hz，其他两个尺度对应的弱信号是小波其他成分与信号相互作用的结果。



a）Daub16小波叠加在频率为20Hz的正弦信号上　　b）信号的CWT中得到充分显现

图 11.25　a) Daub16 小波叠加在频率为 20Hz 的正弦信号上。三个图形中的正弦信号频率都是相同的，但小波信号的尺度伸展到不同的尺度，显示了该频率的正弦信号与不同尺度的小波信号之间的相关性。此在图 b)所示信号的 CWT 中得到充分显现。主频 20Hz 出现在尺度 7 附近，而在上面的两个箭头所指尺度上，也存在微弱的相关性

我们可以将小波的中心频率与尺度对应起来。MATLAB函数centfrq计算出的中心频率假设尺度为1，抽样频率为1。对于任意的尺度 s_c 和抽样频率 f_s，与此尺度相对应的中心频率 $f=f_{ctr}*f_s/s_c$，其中 f_{ctr} 是尺度为1对应的中心频率。在MATLAB中函数scal2frq可以实现这样的转换。在图11.25中20Hz频率对应尺度为6.67。

例11-7　时频小波显示

利用MATLAB函数scal2frq显示信号 $f(t)=\cos(200\pi t)$ 的CWT，体现小波变换的时频特性。采用Morlet小波。

解：

```
%ExmpFreq8.m
wavef = 'morl'; %wavelet type
imap = invertcolormap(colormap(gray(128)));
fs = 1000;T = 1/fs;
t = 0:T:2;
f = cos(200*pi*t); %signal
fr = 10:10:500; %Define the frequencies
%Find the corresponding scale numbers
s = centfrq(wavef, 8)./(fr*T);
CWTcoeffs = cwt(f,s,wavef);
figure(1);clf; %Create a scaled image
imagesc(t, fr, abs(CWTcoeffs));
set(gca,'YDir','normal') %Can be used to invert image
colormap(imap);
haxis = gca; %Get Current Axis handle
colorbar
axisTITL = get(haxis,'title');
axisXLAB = get(haxis,'xlabel');
axisYLAB = get(haxis, 'ylabel');
str = ['CWT using 'wavef ];
set(axisTITL,'String',str);
set(axisXLAB,'String','time in seconds');
set(axisYLAB, 'String', 'approximate frequency in Hz');
```

图11.26显示了结果。

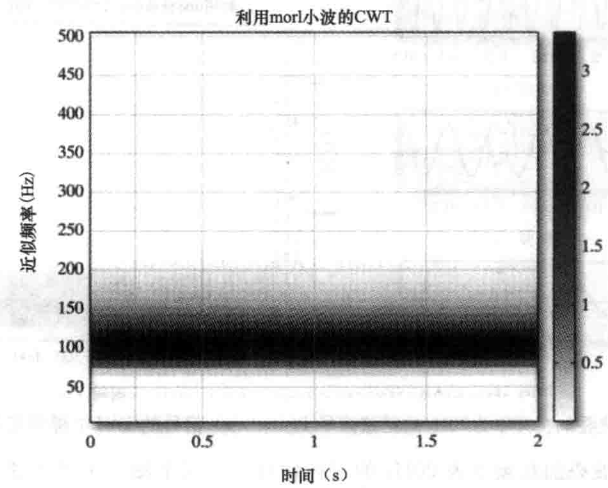


图11.26　频率为100Hz余弦信号的CWT。通过函数scal2frq将纵轴的尺度值转换为近似的频率。函数invertcolormap是自定义函数，参见附录D

时域。小波在时域提取信号中的小事件比较优越。由于小波信号是时域有限的信号，因此该特性是直观的，也是意料之中的。考虑一个持续时间为2s的信号，前1s是正弦信号，后1秒为余弦信号。下列MATLAB代码产生该信号，如图11.27所示。

```
fs = 2000;T = 1/fs;
t = 0:T:2-T;
f1 = sin(2*pi*100*t); %signal at 100 Hz
f2 = cos(2*pi*100*t);
f = [(f1(1:length(t)/2)) (f2(length(t)/2+1:length(t)))];
```

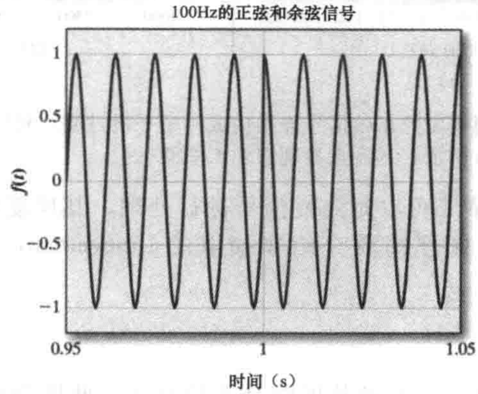


图11.27　一个正弦信号突然在1s处转变为余弦信号。正余弦信号的频率都是100Hz，抽样频率是2kHz

由于信号的持续时间为2s，抽样频率为2000Hz，因此信号的不连续点出现在样点2000处。通过信号的CWT，我们可以清晰地发现此不连续点。下列MATLAB代码完成信号的CWT变换。图11.28显示了三个小波Haar、Daub16和Morlet对应的变换结果。

```
wavef = 'haar'; %wavelet type
CWTcoeffs = cwt(f,1:64,wavef, 'plot');
```

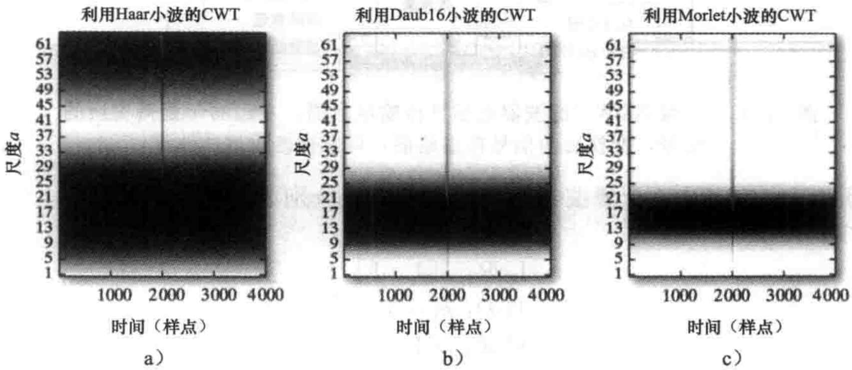


图11.28　图11.27所示正弦/余弦信号，分别对应Haar小波、Daub16小波和Morlet小波的连续小波变换。这三个小波对应的CWT都在时域有效地检测到不连续点

即使对于人眼不易察觉的信号渐变，时频分析也能检测出来。观察两个不同幅度构成的正弦信号，两个信号之间的过渡是平滑的，如图11.29所示。利用墨西哥帽小波可以在样点2000处凸显其过渡位置。这样的过渡在频率上覆盖所有64个尺度。

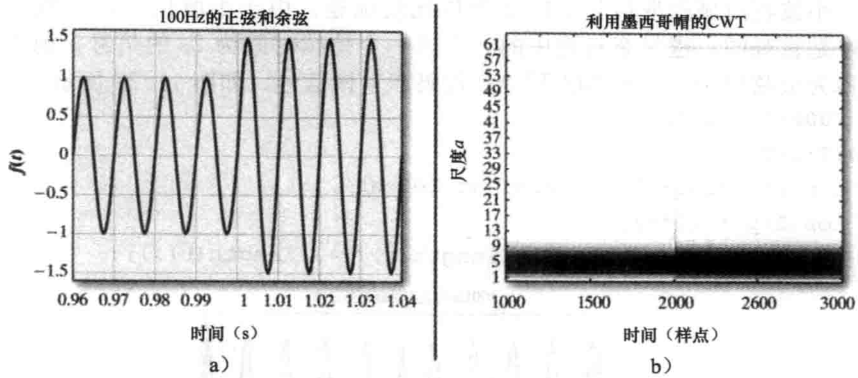


图 11.29　两个同频不同幅度的正弦信号连接构成一个平滑过渡。利用墨西哥帽的连续小波变换结果如图 b 所示，过渡点显现在样点 2000 处

我们已经注意到，CWT 可以对离散信号进行处理，其尺度(频率)变量是连续的。与之相对应的便是 DWT，顺便简要介绍非抽取的(undecimated)离散小波变换(常称为 NDWT)。

11.5　NDWT

NDWT 与 CWT 类似，只不过其展缩是 2 的幂次。此展缩因子称为二分的(dyadic)。第一个尺度因子是 2，下一个是 4，然后是 8，如此递增。而在 CWT 中，尺度因子是 2、3、4、5，依次递增。

我们可以将 NDWT 看作由两个滤波器组成，一个是高通滤波器，另一个是低通滤波器，分别对应母小波和尺度函数。按照传统意义上解释，高通部分产生细节(detail)系数 CA，而低通部分产生近似(approximation)系数 CD，如图 11.30 所示。

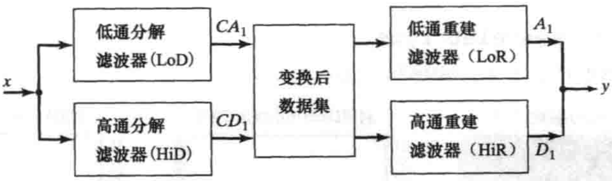


图 11.30　一级 NDWT 滤波器组信息传输示意图。左边的 x 是待变换的信号，变换后的信号在重建前，可能被修改或滤波等

对于 Haar 小波，这 4 个滤波器的单位脉冲响应分别为

$$\text{LoD}: k[1\quad 1]$$
$$\text{LoR}: k[1\quad 1]$$
$$\text{HiD}: k[-1\quad 1]$$
$$\text{HiR}: k[1\quad -1]$$

常量 k 的值为$\sqrt{2}/2$，用来保持信号的能量不变。下面分析该系统的单位脉冲响应，首先输入一个单位脉冲序列至系统的低通部分，我们通过脉冲与 LoD 滤波器的单位脉冲响应卷积得到其输出，再将该输出通过 LoR 滤波器。低通部分的分析过程可表示为

$$CA_1 = [1] * k[1\quad 1] = k[1\quad 1]$$
$$A_1 = k[1\quad 1] * k[1\quad 1] = [1/2\quad 1\quad 1/2]$$

同理，可得高通部分的分析过程。

$$CD_1 = [1] * k[-1\quad 1] = k[-1\quad 1]$$

$$D_1 = k[-1\quad 1] * k[1\quad -1] = [-1/2\quad 1\quad -1/2]$$

最后得到输出 $y = A_1 + D_1 = [0\quad 1\quad 0]$。输出除了平移一个单位外，其与输入相等。

在上述方程式中，输入信号经过 LoD 和 HiD 变换后，变换后的数据集没有做任何处理，输入信号在输出端得到了完全重建。我们也可以修改变换后的数据集，以得到某些期待的特性，如去除噪声，此种情形下，我们按照预先设定来改变变换后数据，从而得到我们所需要的修改版的输入信号。

例 11-8 利用 MATLAB 求解信号的 NDWT

在前面的 NDWT 描述中，利用卷积计算来获得输出和变换系数 CA_1 和 CD_1。此例展现如何利用 MATLAB 获得同样的结果。

解：MATLAB 函数 ndwt 可以用来计算信号的 NDWT。

```
x = 1;
level = 1;
wname = 'haar';
sigExtension = 'zpd'; %zero padding
w = ndwt(x, level, wname, 'mode', sigExtension);
```

函数 ndwt 的输出 w 是一个数据结构，其包含多个数据项。我们感兴趣的是所用的滤波器组(LoD，HiD，LoR，HiR)和变换系数(CA_1，CD_1)。这些数据可以通过下列方式显示。

```
disp(w.filters); %Shows LoD, LoR, HiD, and HiR
disp(w.dec{1}); %Gives approximation coefficients
disp(w.dec{2}); %Gives detail coefficients
```

此例中的结果为

```
LoD: [0.7071 0.7071] %filters
HiD: [-0.7071 0.7071]
LoR: [0.7071 0.7071]
HiR: [0.7071 -0.7071]
0.7071 0.7071 %Reconstruction coefficients
-0.7071 0.7071 %Detail coefficients
```

近似系数和细节系数与我们通过“手动”卷积计算的结果相同。 ◀

在图 11.30 中，LoD 和 LoR 两个滤波器构成一个低通滤波通道，对于 Haar 小波来说，其传输函数(实际是系统的单位脉冲响应)是 LoD 和 LoR 滤波器的脉冲响应之卷积。

$$\text{低通} \rightarrow k[1\quad 1] * k[1\quad 1] = [1/2\quad 1\quad 1/2]$$

高通滤波通道由 HiD 和 HiR 两个滤波器构成，其脉冲响应是该两个滤波器的脉冲响应之卷积。

$$\text{高通} \rightarrow k[-1\quad 1] * k[1\quad -1] = [-1/2\quad 1\quad -1/2]$$

此两个单位脉冲响应对应的系统频率响应如图 11.31所示，其坐标轴为归一化坐标轴。

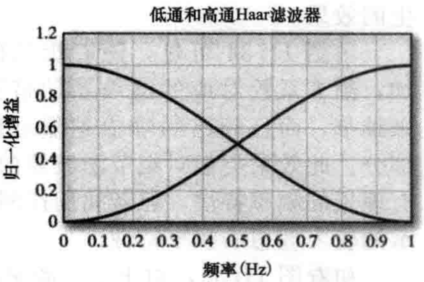


图 11.31 LoD 与 LoR 卷积以及 HiD 与 HiR 卷积分别构成的低通和高通半带滤波器

下一级。为了得到第二级，我们对第一级输出的近似系数 CA_1 进行相同的滤波处理，图 11.32 显示了该结构。系数 CA_2 由输入信号通过两次 LoD 滤波得到，系数 CD_2 由输入信号通过 LoD 和 HiD 滤波得到。输出的三组系数 CA_2、CD_2 和 CD_1 对应的等效滤波器为

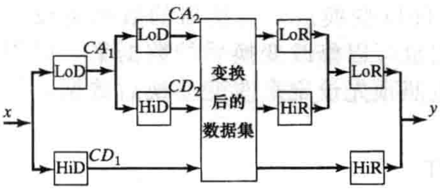
CA_2 对应的等效滤波器是 LoD→

$$\text{LoD}: k[1\ 1] * k[1\ 1] = [1/2\quad 1\quad 1/2] \tag{11.8}$$

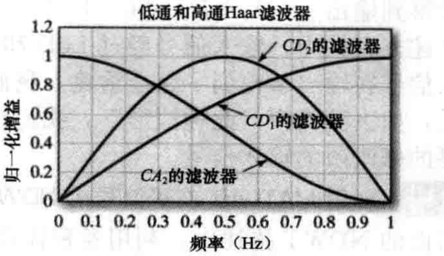
CD_2 对应的等效滤波器是 LoD→

$$\text{HiD}: k[1\ 1] * k[-1\ 1] = [-1/2\quad 0\quad 1/2] \tag{11.9}$$

$$CD_1 \text{ 对应的等效滤波器是 HiD}:k[-1 \quad 1] \tag{11.10}$$



a）一个二级NDWT框图，近似系数受进一步滤波和重构的影响



b）各变换系数对应的等效滤波器的频率响应

图 11.32　二级 NDWT 的框图及各变换系数对应滤波器的频率响应

例 11-9　MATLAB 和 Haar 滤波器

利用 MATLAB 可以得到与上述二级 Haar 滤波器组相同的结果，Haar 滤波器表示方法见式(11.7)、式(11.8)和式(11.9)。

解：我们利用非抽取离散小波变换函数 ndwt，如下列代码所示：

```
x = 1;                                            %impulse input
level = 2;
wname = 'haar';
w = ndwt(x, level, wname, 'mode', 'zpd'); %zero padding
disp(w.dec{1});                                   %CA2
disp(w.dec{2});                                   %CD2
disp(w.dec{3});                                   %CD1
```

上述代码输出以下系数：

```
 0.5000     1.0000     0.5000                     %CA2 coefficients
-0.5000     0          0.5000                     %CD2 coefficients
-0.7071     0.7071                                %CD1 coefficients
```

按照相同的方式，分解可以进一步应用到更多级，小波函数也可以是其他小波系列。图 11.33 显示了一个拓展到 4 级的非抽取离散小波变换。为了观察等效滤波器，我们绘出了这些逐级增加所对应的等效滤波器的频率响应。图 11.34 显示了 Haar 小波产生的效果。

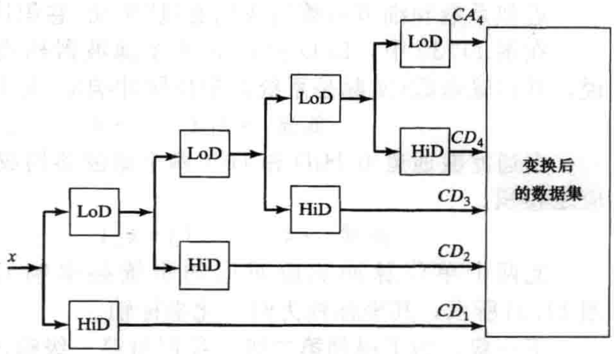


图 11.33　一个 4 级非抽取的 DWT 框图

由图 11.34 可见，随着级数的增加，细节系数对应的滤波器的带宽越来越窄，而且频谱的峰值越靠近低频部分。此效果类似对细节滤波器的脉冲响应进行过抽样(oversampling)。Haar 小波远不具有理想的频率特性，但若我们按照因子 2 过抽样其细节滤波器，将会出现类似图 11.31 所示的较为理想的频率特性。

初看图 11.35，由于信号频率越高的部分可以具有较低的频率分辨率，因此整个频率分布图感觉应该是左右调换。实际并非如此。对于许多实际的信号，主要信号大多包含在信号的低频部分，而高频部分主要反映信号的细微差别和波动。噪声也常常出现在信号的高频分量。

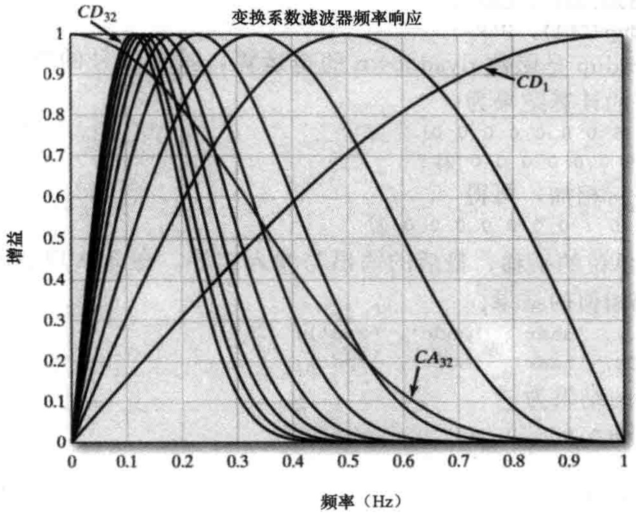


图 11.34 此图显示了一个利用 Haar 小波从 1 至 32 级所对应的等效的系数滤波器频率响应

我们可以将非抽取离散小波变换看作由一系列内插(up-sampled)的细节滤波器组成，另一种可实现相同结果的方式就是抽取(down-sample)信号，这就是 DWT。

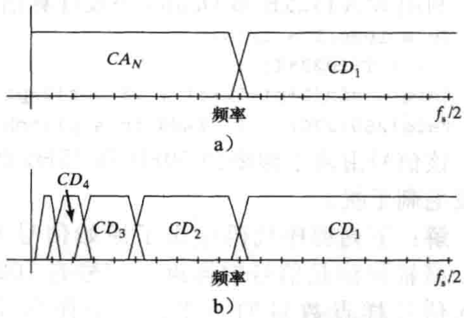


图 11.35 理想化的系数滤波器

11.6 DWT

DWT 非常类似于非抽取 DWT，DWT 通过信号的抽取来代替滤波器的延展。DWT 的框图如图 11.36 所示。

例 11-10 单位脉冲与 DWT

利用 Haar 小波展现单位脉冲信号通过一级 DWT 的过程。利用 MATLAB 函数 dwt 和 idwt 验证其结果。

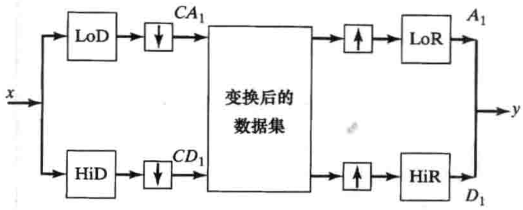


图 11.36 一级 DWT 的框图。含有上箭头的方框表示信号的 2 倍抽取，含有下箭头的方框表示信号的 2 倍内插。

解： 对于 Haar 小波，对应的 4 个滤波器具有下列单位脉冲响应

```
LoD = [1 1];
HiD = [-1 1];
LoR = [1 1];
HiR = [1 -1];
```

假如输入信号为

```
x = [1 0 0 0 0 0 0 0];
```

可以计算 CA_1 和 CD_1

```
CD1 = dyaddown(conv(x, HiD));
CA1 = dyaddown(conv(x, LoD));
```

其中，dyaddown 是 MATLAB 函数，其通过仅保留信号中的偶数项而实现对信号的 2 倍抽取。CA_1 和 CD_1 的计算结果如下：

```
CD1 = [0.70711 0 0 0]
CA1 = [0.70711 0 0 0]
```

假如不对这些输出进行修改，那么可以利用下列程序恢复信号 A_1 和 D_1：

```
A1 = conv(dyadup(CA1), LoR);
D1 = conv(dyadup(CD1), HiR);
```

其中，函数 dyadup 是函数 dyaddown 的逆运算，其在信号的奇数项上插入零值的样点。信号 A_1 和 D_1 的计算结果为

```
D1 =[0 0.5 -0.5 0 0 0 0 0 0 0]
A1 =[0 0.5 0.5 0 0 0 0 0 0 0]
```

将 A_1 和 D_1 进行相加，可得

```
y = A1 + D1 = [0 1 0 0 0 0 0 0 0 0]
```

除了出现一个单位的平移，最后的输出与输入相等。在 MATLAB 中，可以利用函数 dwt 和 idwt 来得到相似的结果。

```
[CA CD] = dwt(x, 'haar', 'mode', 'zpd');
y = idwt(CA, CD, 'haar', 'mode', 'zpd');
```

CA、*CD* 和 *y* 的结果为

```
CD1 = 0.70711 0 0 0
CA1 = 0.70711 0 0 0
y = 1 0 0 0 0 0 0 0
```

◀

例 11-11　利用 MATLAB 和 Daub4 小波计算 DWT

利用 MATLAB 和 Daub4 小波计算信号的一级 DWT 的系数，信号定义如下：

```
fs = 1000;T = 1/fs;
t = 0:T:1023*T;
fsig = sin(2*pi*50*t) + .5*cos(2*pi*75*t);
fsig(250:270) = 1; %Add in a glitch
```

该信号由两个频率为 50Hz 和 75Hz 的正弦信号组成，中间在样点 250～270 之间插入一段毛刺干扰。

解： 下列程序代码绘出了原始信号 fsig 的波形、近似系数 *CA* 和细节系数 *CD*。注意，横轴标注是信号的样点，信号有 1024 个样点。在抽取后，输出系数 *CA* 和 *CD* 只有输入信号样点数目的一半。三个图形中的横轴时间范围都是 0～1023*T* 秒。结果如图 11.37所示。

```
wavef = 'db2';   %Daub4
[CA1 CD1] = dwt(fsig,wavef);
subplot(3, 1, 1);
plot(t/T, fsig);
%
subplot(3, 1, 2);
plot(CA1);
%
subplot(3, 1, 3);

plot(CD1);
```

a)

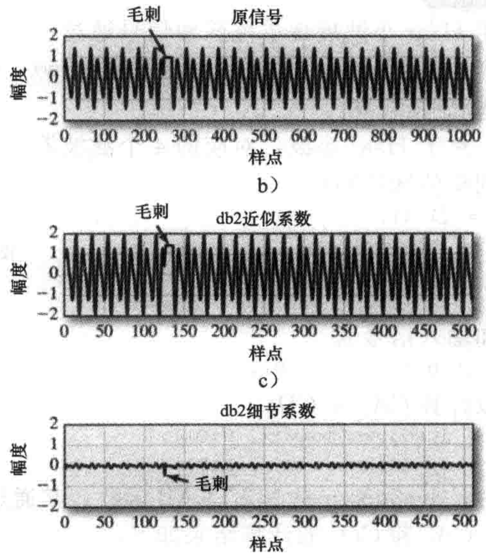


图 11.37　利用 dwt 函数来计算具有一段毛刺的正弦信号的一级系数　◀

混叠。在图11.36所示的DWT结构中，当我们进行抽取时，需要考虑可能存在的混叠问题。在低通通道，并不会出现混叠问题，因为2倍抽取信号前，半带低通滤波器滤除了可能造成混叠的高频成分。在高通通道，高通滤波器保留了高频分量而滤除了低频分量，在随后的2倍抽取信号中，上半频带的高频分量将与下半频带的低频分量发生混叠。例如，考虑一个由频率100Hz和300Hz构成的正弦信号，对其抽样频率为1000Hz，如图11.38a所示。将该信号通过HiD高通滤波器，得到的输出信号的频谱如图11.38b所示。信号的2倍抽取相当于将信号的抽样频率降低到原来的一半(即500Hz)，造成了上半频带的频率混叠到下半频带。由于下半频带在混叠位置区域是空的，因而此混叠过程没有造成信号的信息损失，如图11.38c所示。在图11.38d中，内抽提高抽样频率至原来的数值(即1000Hz)，在上半频带中出现了下半频带的一个镜像。最后，在图11.38e中，高通半带HiR重建滤波器滤除了下半频带，我们得到了原来信号的上半频带。同理，低通通道最终可以得到原来信号的下半频带。两路信号相加，即可恢复得到原来的信号。

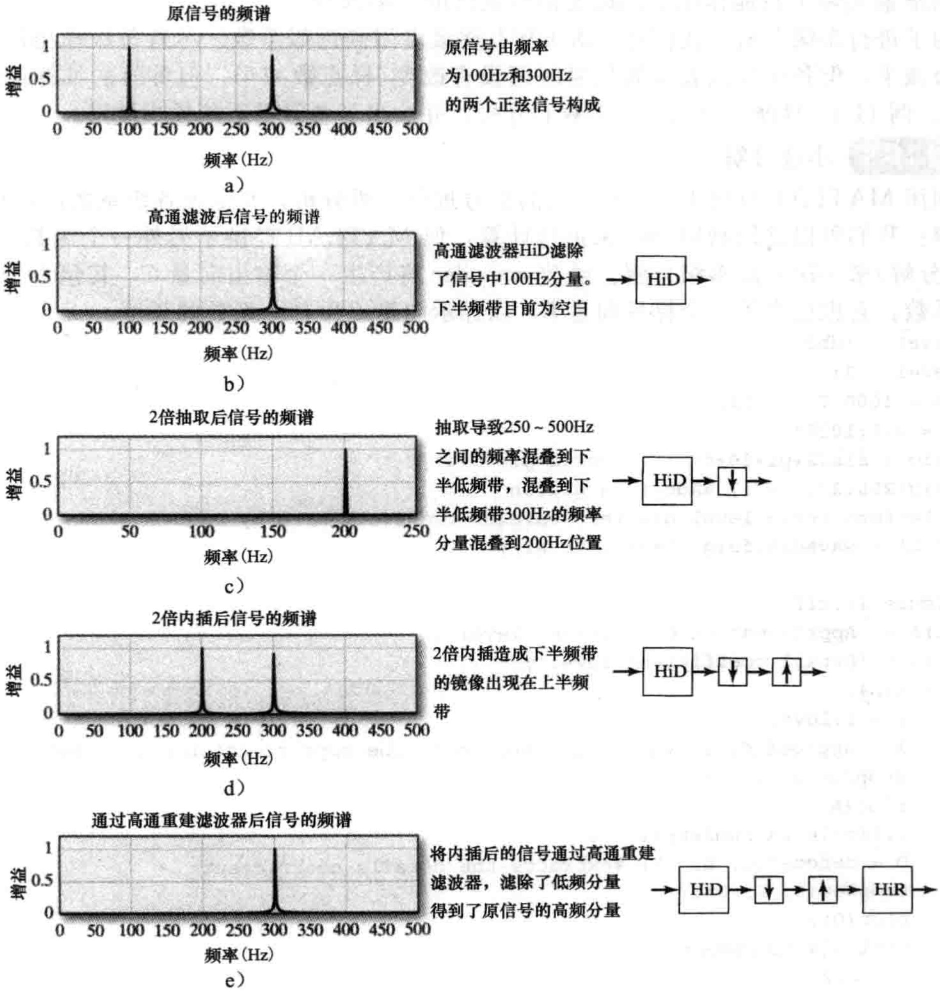


图11.38 一个由频率100Hz和300Hz构成的正弦信号，先通过一个HiD高通滤波器，然后2倍抽取，再2倍内插，最后通过HiR高通滤波器，没有出现信号的损失

多级。与非抽取DWT的情形一样，图11.36中的结构框图很容易拓展到多级。

图 11.39显示了三级 DWT 对应的系数。由于数据是离散的，我们逐级按 2 倍抽取，直至信号分解到需要的级数。例如，在图 11.39 中，假如信号原始具有 1024 个样点，则系数 CD_1 和 CA_1 将分别具有 512 个样点，系数 CD_2 和 CA_2 将分别具有 128 个样点。

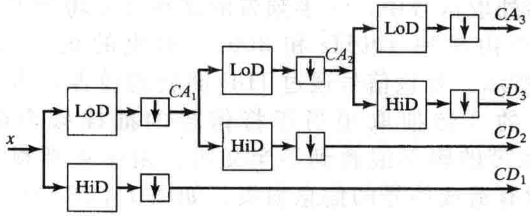


图 11.39 产生变换系数的 3 级 DWT

随着样点数逐级减半，频带宽度也相应地发生同样的改变。每次 2 倍抽取信号，信号的抽样频率相应地减低一半。由于时域的压缩导致频域的扩展，后面的高通 HiD 和低通 LoD 滤波器实际上只能作用于抽取前信号频谱的一半部分。

为了进行多级分析，我们将一级 DWT 逐级应用到近似系数。尽管每级输出的样点数量逐级减半，但各级样点表示的信号时间没有改变(样点数减半，但等效的抽样间隔增加一倍)。例 11-12 解释一个三级的 DWT 分析，并绘出了各级系数的输出波形。

例 11-12 小波分解

利用 MATLAB 对例 11-11 中定义的信号进行三级分析，并绘出各级系数的波形。

解： 我们可以逐级利用 dwt 来进行计算，但 MATLAB 提供了另外一个函数 wavedec (小波分解)来一次完成多级分解。函数 wavedec 将产生一个输出向量 ***C***，其包含了各级的输出系数。它也包含了一个标签向量 ***L***，以标示向量 ***C*** 中各个系数的长度。

```
wavef = 'db2';
level = 3;
fs = 1000;T = 1/fs;
t = 0:T:1023*T;
fsig = sin(2*pi*50*t) + .5*cos(2*pi*75*t);
fsig(250:270) = 1; %Add in a glitch
% Perform three-level discrete wavelet transform of s by Daub4.
[C L] = wavedec(fsig, level, wavef);
%
figure(1);clf;
strA = 'Approximation coefficient level';
strD = 'Detail coefficient level';
n = 1024;
for i = 1:level
    A = appcoef(C, L, wavef, i); %extracts the approximation coefficients
    subplot(3, 2, 2*i-1)
    plot(A);
    title([strA num2str(i)]);
    D = detcoef(C, L, i); %extracts the details coefficients
    subplot(3, 2, 2*i)
    plot(D);
    title([strD num2str(i)]);
    n = n/2;
end
```

该程序的输出如图 11.40 所示。再次强调，尽管从图形上看起来各级似乎是相同的样点，但实际上样点数量逐级减半。6 个图形的时间尺度是相同的，因此那一小段毛刺出现在所有图形中相同时间的位置。

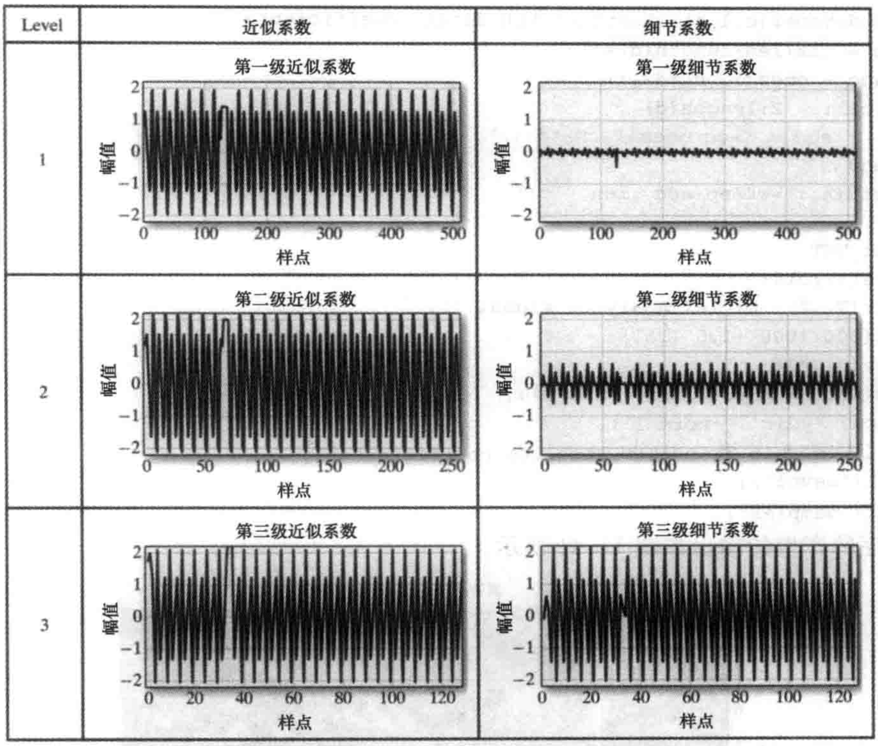


图 11.40　含有毛刺的正弦信号三级 DWT，各级的近似系数在左边，各级的细节系数在右边

为得到一个传统的 DWT 输出，我们利用颜色或灰度逐级表现系数随时间分布情况。在显示过程中有必要根据各级样点数相应调整时间轴的比例，具体的做法就是扩展每个样点的时间宽度。例如，假如我们有一个位于三级的系数[1 2 3 4]，我们可以通过加宽每个样点至原来的 2 倍来改变其为长度为 8，得到序列[1 1 2 2 3 3 4 4]。例 11-13 就是这样一个 8 级小波分解过程的显示。

例 11-13　含有毛刺信号的 DWT

利用 Daub4 小波和 MATLAB 产生下列定义信号的各级 DWT 随时间的分布图

```
fs = 1000;T = 1/fs;
t = 0:T:4095*T;
f = sin(2*pi*100*t); %create the signal
f(894:903) = 1;      %Add a glitch
```

该信号 DWT 的显示应为 8 级。根据输入信号的特点解释信号 DWT 的数据。

解：

```
% A traditional plot of an 8 level dwt.
wavef = 'db2';s = 8;  %Daub4 with 8 levels
fs = 1000;T = 1/fs;
t = 0:T:4095*T;
f = sin(2*pi*100*t);  %create the signal
f(894:903) = 1;       %Add a glitch
%Do DWT to 8 levels
[c l] = wavedec(f, s, wavef);
len = length(f);
cfd=zeros(s,len);     %detail coefficient matrix
%This loop makes the detail coefficents the same length = len.
%It does this by duplicating values where needed.
for k=1:s
```

```
        d=detcoef(c,l,k); %extract kth detail coefficients
        n = fix(len/length(d)+1);
        acc = ones(1, n)*d(1);
        for i = 2:length(d)
            acc = [acc ones(1, n)*d(i)];
        end
        cfd(k,:)=wkeep(acc, len, 'c');
    end
    % Plot DWT.
    figure(1);clf;
    plot(t/T, f); title('Analyzed signal.');
    axis([500 1000 -1.5 1.5]);
    figure(2);clf;
    image(wcodemat(cfd, 255, 'row'));;
    set(gca, 'ydir', 'normal');
    title('Discrete Transform,absolute coefficients');
    ylabel('Level');
    xlabel('samples');
```

该程序的产生的输出如图 11.41 所示。

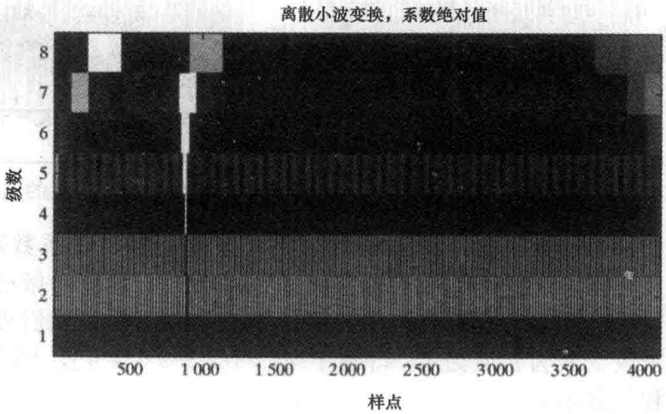


图 11.41　含有毛刺正弦信号的 8 级 Daub4 小波分析，横轴为时间(样点)，纵轴为级数

在图 11.41 中，在左上部和右边角都出现了某些边界效应。毛刺干扰位置大约在样点 900 处清晰可见。第一级的输出是信号通过高通滤波器 HiD 的输出，该输出信号的频谱保留了输入信号的上半频带部分的频谱。由于抽样频率 f_s 为 1000Hz，因而第一级输出对应输入信号的 250～500Hz 频率范围。其他各级输出对应的频率范围大致如表 11.2所示。在图 11.41 中，第二级最亮，因其包含了 100Hz 的输入信号。干扰穿过各级，并且漂亮地出现在样点 900 处。

表 11.2　图 11.41 中各级表示的频带。假设各级滤波器近似为理想滤波器

级数	大致的频率范围(Hz)
1	250～500
2	125～250
3	62.5～125
4	31.3～62.5
5	15.6～31.3
6	7.8～15.6
7	3.9～7.8
8	1.95～3.9

简单噪声滤除——阈值化。多级小波分解将信号中的不同频率分量分离出来，在根据这些分量重建信号之前，我们可以分别对这些不同分量进行相应地处理。对于最简单的情形，我们可以将某些频率范围的分量设置为零，从而将其从信号中滤除。对于稍微复杂些的处理，我们可以设置一个阈值，对于各级小波变换系数进行阈值化处理，当系数值大于阈值时，将其置为零，或者根据某些场合将小于阈值的系数置为零。由于小波分析可以为信号提供较好的时间分辨率(同时具有时间分辨率和频率分辨率)，因此我们可以对很窄的一段信号的某个频带进行阈值化处理。下列的例题可以阐述这些概念。

例 11-14 利用 DWT 实现信号重建

考虑下面给定的信号

$$f(t) = t^5 - 12.4t^4 + 53.61t^3 - 95.054t^2 + 61.814t - 8.97$$

对其 0～4.095s 的一段信号按频率 1000Hz 进行抽样，再叠加一个频率为 400Hz 的正弦噪声信号。利用 Daub4 小波对此含有噪声的信号进行 4 级 DWT，然后逐级让细节系数为零重建信号，显示其对应的结果。

解：

```
x = [1 -12.4 53.61 -95.054 61.814 -8.97];
fs = 1000;T = 1/fs;
t = 0:T:4095*T;
f = 0;N = length(x);
for i = 1:N
    f = f + x(i)*t.^(N - i);
end
%normalize and add in the noise
mx = max(f);mn = min(f);
f = (f-mn)/(mx-mn);
fn = f + .03*sin(2*pi*400*t);
```

下面利用小波分解得到一个 4 级小波系数。

```
wavef = 'db2';
s = 4; %level
[C L] = wavedec(fn, s, wavef);
[d1 d2 d3 d4] = detcoef(C, L, 1:4);
a4 = appcoef(C, L, wavef, 4);
```

在上述代码中，d_1 表示第一级细节系数向量。我们将其设置为零，然后重建信号，相应的代码如下：

```
d1 = d1*0;                    %vector of zeros
C = [a4 d4 d3 d2 d1];         %reform C
fr = waverec(C, L, wavef);%Reconstruct signal
subplot(6, 1, 6);
plot(t, fr);
```

我们可以对其他级的细节系数向量进行同样的处理。结果如图 11.42 和图 11.43 所示。

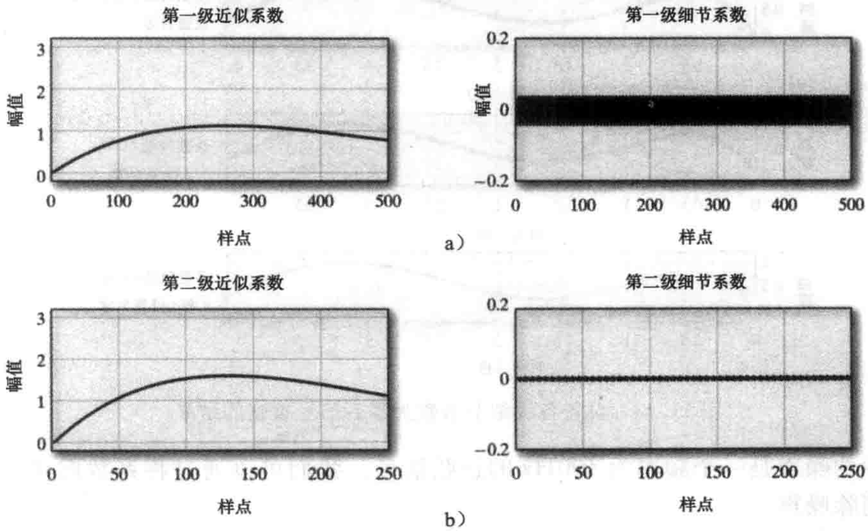


图 11.42 小波系数。各级的近似系数在左边，细节系数在右边。第一级的细节系数的数值最大，因为这级对应信号中的噪声

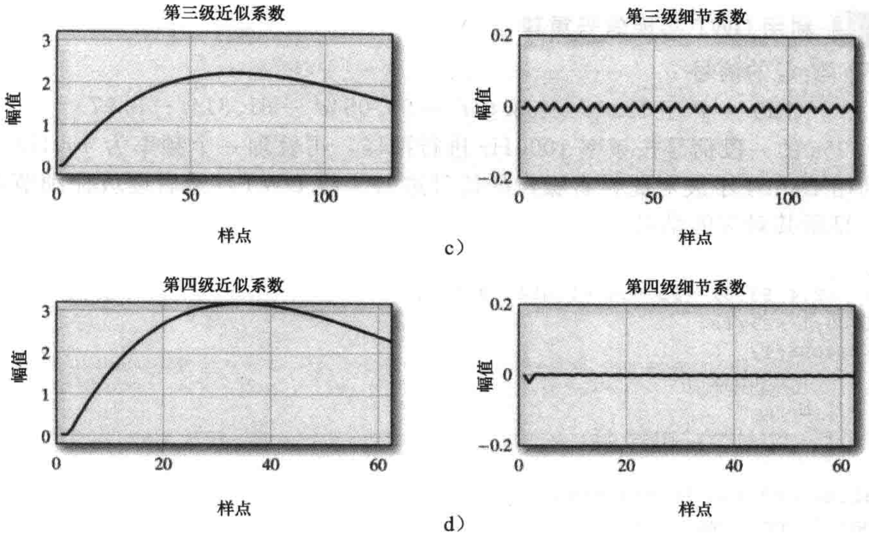


图 11.42 （续）

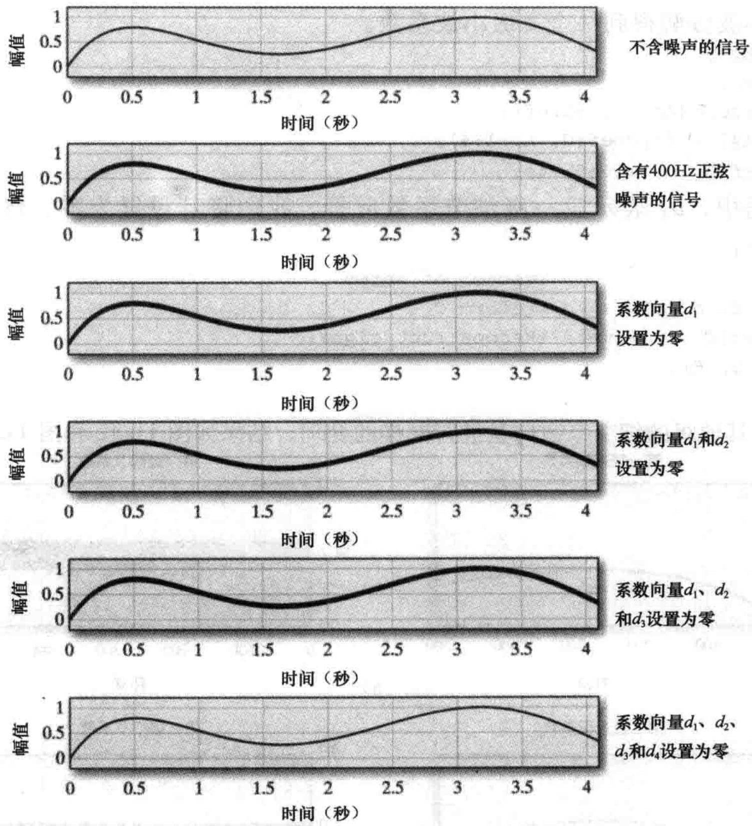


图 11.43　设置各级细节系数为零后信号重建的结果

信号中的噪声是一个频率为 400Hz 的正弦信号。我们可以通过将系数向量 $d_1 \sim d_4$ 设置为零来消除噪声。◀

阈值化方法主要有两种。第一种阈值化称为硬阈值(hard threshgholding)，也是最直观的方法。在硬阈值化处理中，位于阈值设定范围的信号项置为零，否则信号保持不变。

假如阈值设置得比较小，则这种阈值处理几乎没有意义。假如阈值设置得比较大，则可能引起某些难以预料的跳跃。第二种阈值化称为软阈值(soft threshholding)。在软阈值化处理中，将信号绝对值中小于设定阈值的项置为零，其他项都减去阈值。

考虑信号 x，其由两个数值组成 $x=[0.4\ 0.6]$。假如设置阈值 Th=0.5，则对信号进行硬阈值的结果为[0 0.6]。而在软阈值情况，第一个数值 0.4 小于阈值，因此被设置为零。第二个数值 0.6 高于阈值，因此其值减去阈值，最后得到[0 0.1]。由此可见，当阈值较高时，软阈值不会产生从零的跳变。

例 11-15　小波阈值化

此例中我们观察一个被白噪声信号污染的斜坡信号，信号定义为

```
fs = 10000;T = 1/fs;
t = 0:T:4095*T;
fsig = 5*[t(1:3000) t(1:1096)];
%add random noise uniform distribution
a = -.2;b = .2;
r = a + (b-a).*rand(1, 4096);
fn = fsig + r;
```

利用 Daub4 小波对信号进行 6 级分析。利用阈值化方法处理细节系数以抑制噪声。图 11.44显示了信号的波形。

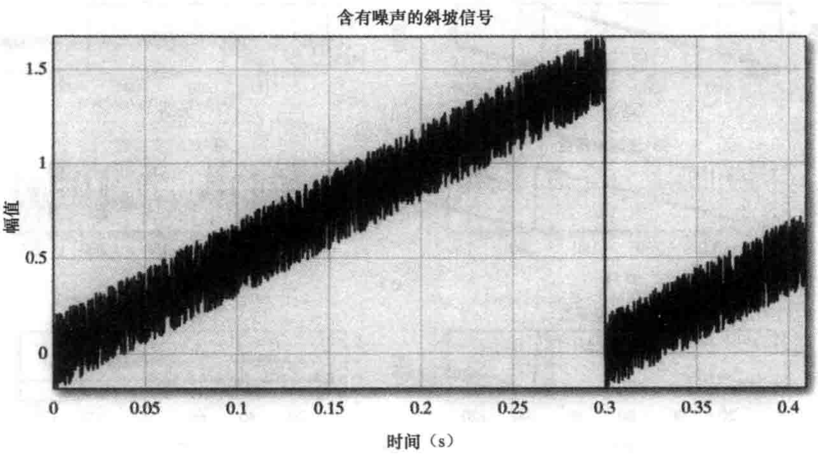


图 11.44　叠加噪声的斜坡信号

解：我们首先进行信号的小波分析，然后绘出其近似系数和细节系数。MATLAB 程序代码如下，计算结果如图 11.45 所示。

```
s = 6;            %level
wavef = 'db2';
dwtmode('sym');
[C L] = wavedec(fn, s, wavef);
figure(1);clf;
strA = 'Approx coeff level';
strD = 'Detail coeff level';
for i = 1:s
    A = appcoef(C, L, wavef, i);
    subplot(s, 2, 2*i-1)
    plot(A);
    title([strA num2str(i)]);
    xlabel('samples');
    ylabel('amplitude');
```

```
    axis tight;
    D = detcoef(C, L, i);
    subplot(s, 2, 2*i)
    plot(D);
    title([strD num2str(i)]);
    xlabel('samples');
    ylabel('amplitude');
    axis tight;
end
```

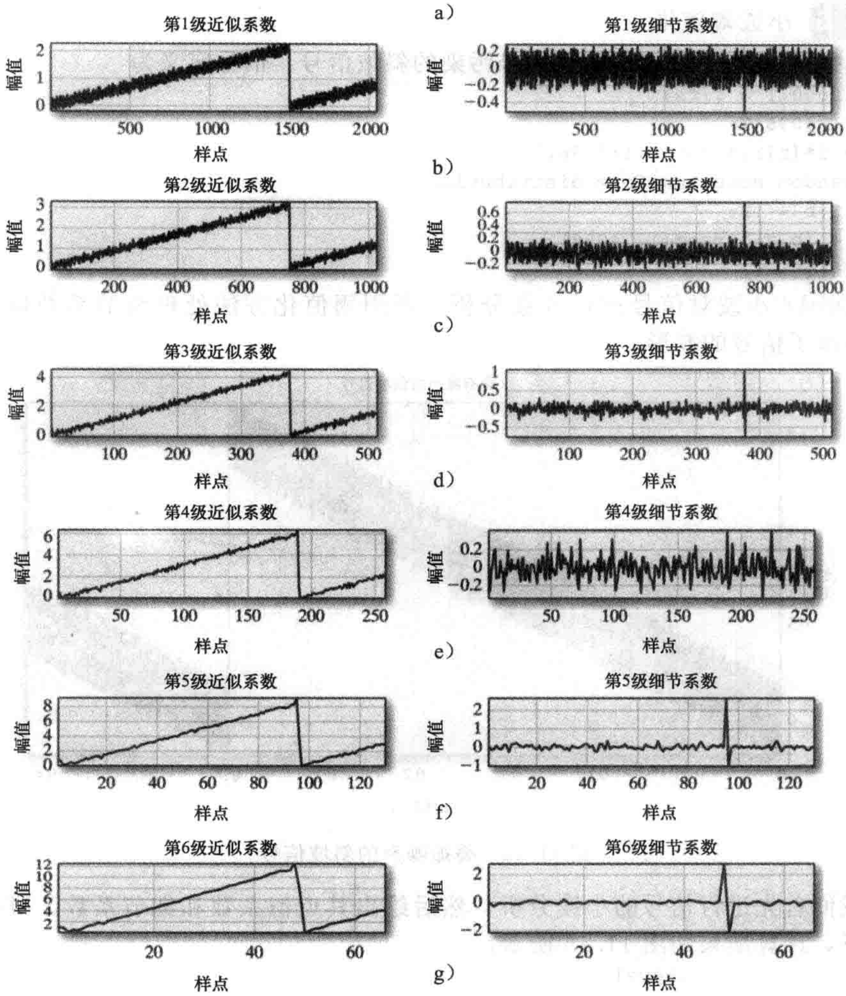


图 11.45 利用 Daub4 小波对信号进行 6 级分解

利用 Daub4 小波对信号进行 6 级分解，可以得到以下 6 个频带的信号分量：1(2500～5000Hz)、2(1250～2500Hz)、3(625～1250Hz)、4(312.5～625Hz)、5(156.25～312.5Hz)和 6(78.125～156.25Hz)。由图 11.45 可见，高频带 1 和 2 包含了最多的噪声，低频带 5 和 6 包含最少的噪声。我们可以大致估算不同频带的噪声量级，频带 1 和 2 低于 0.3，频带 3 和 4 低于 0.2，频带 5 和 6 低于 0.1。我们将利用这些值作为相应频带的阈值。

我们通过这个例子来诠释阈值的效果。我们首先对频带 1 进行阈值处理，然后对频带 1 和频带 2，直至到所有 6 个频带。每次阈值化处理后，都相应地重建阈值化后的信号。相关的程序代码和运行结果如图 11.46 所示。

图 11.46b 显示了对频带 1 进行软阈值后的重建信号。图 c 显示了对频带 1 和 2 进行软阈值后的重建信号。最后，图 g 显示了对所有 6 个频带进行软阈值后的重建信号。

```
s = 6;
wavef = 'db2';
[C L] = wavedec(fn, s, wavef);
figure(2);clf;
D = cell(1, s);
D = detcoef(C, L, 1:s);
a6 = appcoef(C, L, wavef, s);
Thr = [.3 .3 .2 .2 .1 .1];
for i = 1:s
  D{i} = wthresh(D{i}, 's', Thr(i));
  Cr = [a6 D{6} D{5} D{4} D{3} D{2} D{1}];
  fr = waverec(Cr, L, wavef);
  subplot(6, 1, i)
  plot(t, fr, 'b');
  axis tight
  xlabel('time in seconds');
  ylabel('amplitude');
  title('Reconstruction');
end
```

a)

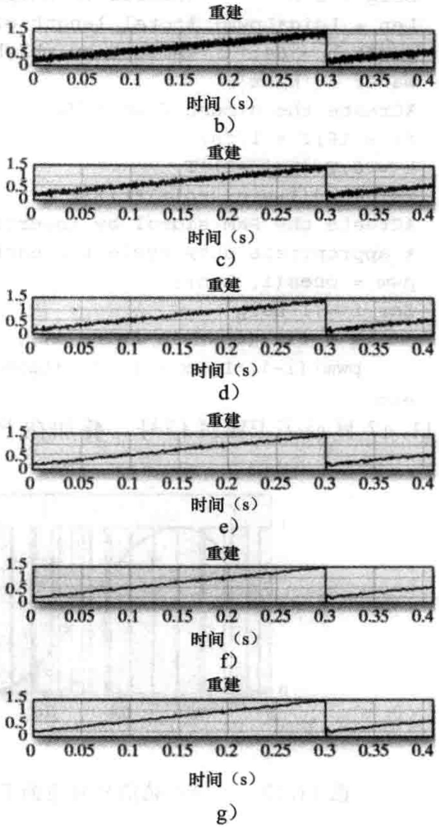


图 11.46 阈值化后信号的重建，图 b 显示在第 1 级阈值化后的重建，图 c 显示在第 1 级和第 2 级阈值化的重建，图 g 显示在第 1 级～第 6 级阈值化后的重建

阈值化处理过程确实明显地抑制了信号中的噪声。

选择恰当的阈值是至关重要的。我们设置的阈值需要大于噪声对应系数的最大值。但是我们也不能将阈值设置得太大，这将导致信号中有用信号的系数损失。如果噪声是方差为 σ^2 的高斯噪声，可以证明噪声系数极可能低于 $\mathrm{Th}=\sigma\sqrt{2\ln(N)}$，其中，$N$ 为样点数[8]。根据此公式计算出的阈值可能比实际需要的阈值要高许多，此阈值可以向下调整。MATLAB 提供了这样的函数 ddencomp，其实现了多个算法以估算一个优化的阈值。◀

例 11-16 含有噪声的 PWM 信号

频率为 1Hz 的正弦信号持续两个周期长度，对其抽样频率为 $16\mathrm{Hz}=2^4\mathrm{Hz}$。对该信号进行 PWM，其中 PWM 信号的基频为 $2048\mathrm{Hz}=2^{11}\mathrm{Hz}$。为了模拟高频噪声，我们以频率 $131072\mathrm{Hz}=2^{17}\mathrm{Hz}$ 加入随机数到信号中，时间位置大约在 0.1s 处，持续时间大约 0.1s。随机数的幅度范围介于 −0.5～0.5 之间。利用 Haar 小波函数对此信号进行 8 级小波分解，绘出近似系数和细节系数。对该短时间内的噪声应用阈值处理后，重建该信号。

解：我们先构造一个抽样频率为 16Hz 的正弦信号。为了产生 PWM 信号，利用 $2^{11}/2^4=128$ 个 PWM 样点块替换正弦信号的每个样点，每个 PWM 样点块的占空比正比于该点正弦信号样点值。为了加入噪声样点，按照因子 $2^{17}/2^{11}=64$ 扩展每个 PWM 样点，

然后在适当的时间间隔叠加噪声。下列MATLAB代码产生正弦信号和PWM信号。

```
Lpwm = 2^7;         %Base frequency for pwm
Lsig = 2^5;         %Number of samples in the signal
Len = Lsig*Lpwm; %total length of pwm signal
stretch = 64;       %Amount by which PWM signal is stretched.
wavef = 'haar';
%Create the sinusoid at 1 Hz
fs = 16;T = 1/fs;
t = 0:T:(Lsig-1)*T;
fsine = (1+sin(2*pi*1*t))/2;
%Create the PWM signal by inserting a PWM cycle with the
% appropriate duty cycle for each sample of the sinusoid.
pwm = ones(1, Len);
for i = 1:Lsig
    k = fix((Lpwm-1)*fsine(i));
    pwm((i-1)*Lpwm + 1 :(i)*Lpwm) = [ones(1, k) zeros(1, Lpwm - k)];
end
```

图 11.47 显示了 PWM 信号，叠加在 PWM 信号上的为正弦信号。

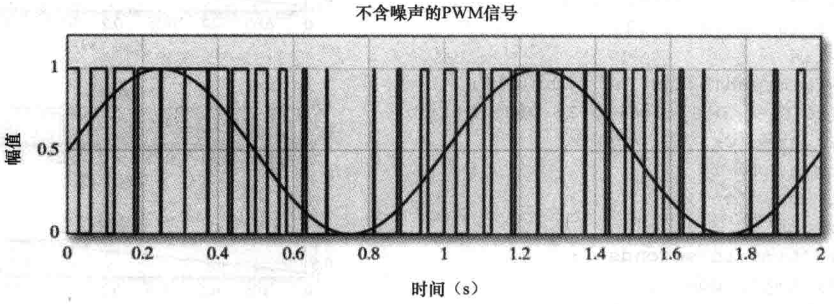


图 11.47　一个正弦信号对应的 PWM 信号。该正弦信号叠加在此 PWM 信号上面

为了加入噪声，以因子 64 对 PWM 信号进行扩展，即将信号中的每个 1 变为 64 个 1，将信号中的每个 0 变为 64 个 0。然后在大约 0.1s 处加入噪声。下列 MATLAB 代码产生变量 pwmstr，并加入噪声。

```
pwmstr = zeros(1, stretch*Len);
for i = 1:length(pwm)
  pwmstr((i-1)*stretch+1:i*stretch) = [ones(1,stretch)*pwm(i)];
end
%add random noise uniform distribution
Len = stretch *Len;
a = -.5; b = .5;
r = a + (b-a).*rand(1, Len);
r(1:10000) = 0;
r(12000: 15000) = 0;
r(20000: Len) = 0;
fn = pwmstr+r;                  %fn is the noisy pwm signal
```

图 11.48 显示了含有噪声的扩展 PWM 信号。此时 x 轴不再表示时间而表示样点。

利用 MATLAB 函数 detcoef 可以一次性得到所有的细节系数，将计算结果存入 cell 中(MATLAB 中 cell 是一个向量数组)。下列代码将细节系数存入 cell 变量 D 中。

```
s = 8; %level
[C L] = wavedec(fn, s, wavef);
D = cell(1, s);
D = detcoef(C, L, 1:s);
```

近似系数也同样可以存入 cell 中，但这里利用循环来分别得到每个近似系数向量。

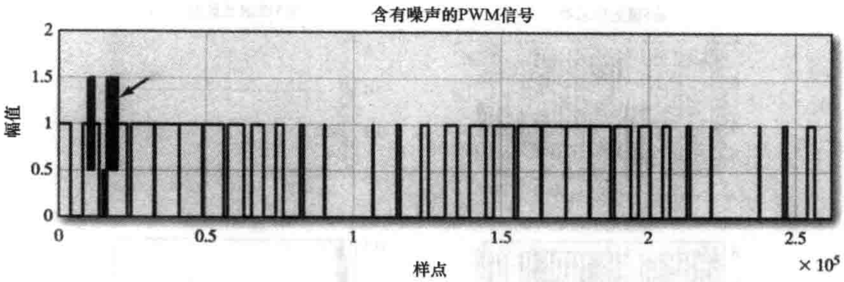


图 11.48 扩展的含噪 PWM 信号

```
A = cell(1, s);
for i = 1:s
    A{i} = appcoef(C, L, wavef, i);
End
```

从 cell 中输出第 i 个系数向量的语法很简便。

```
plot(A{i});
plot(D{i});
```

图 11.49 显示了 8 级小波分解对应的近似系数和细节系数。注意，前 6 个细节系数除了噪声外都等于零。这是因为 Haar 小波函数与二值的 PWM 信号完全匹配的结果。我们

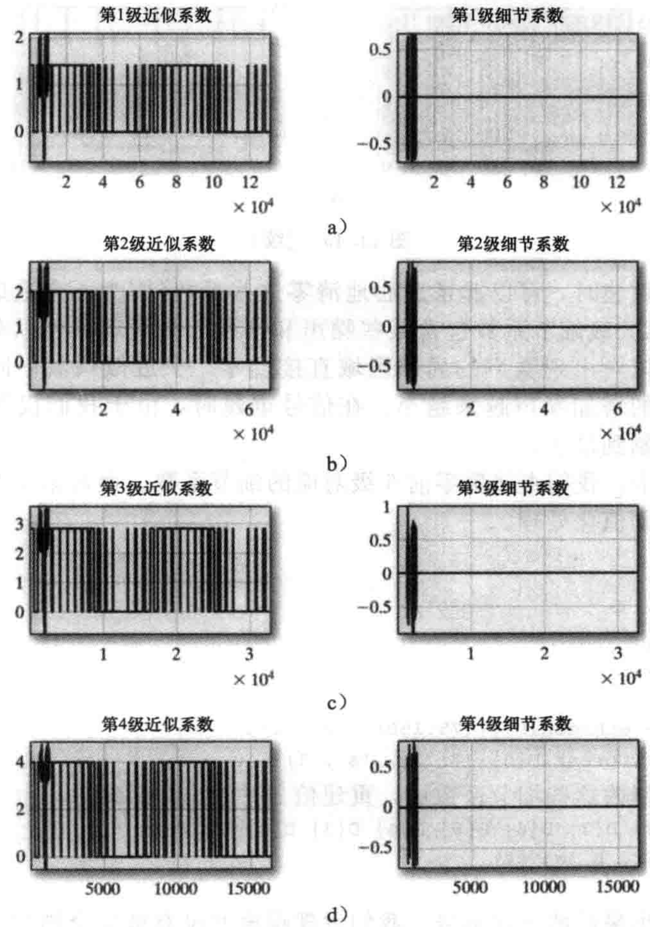


图 11.49 含有噪声的 PWM 信号 8 级小波分解对应的近似系数和细节系数

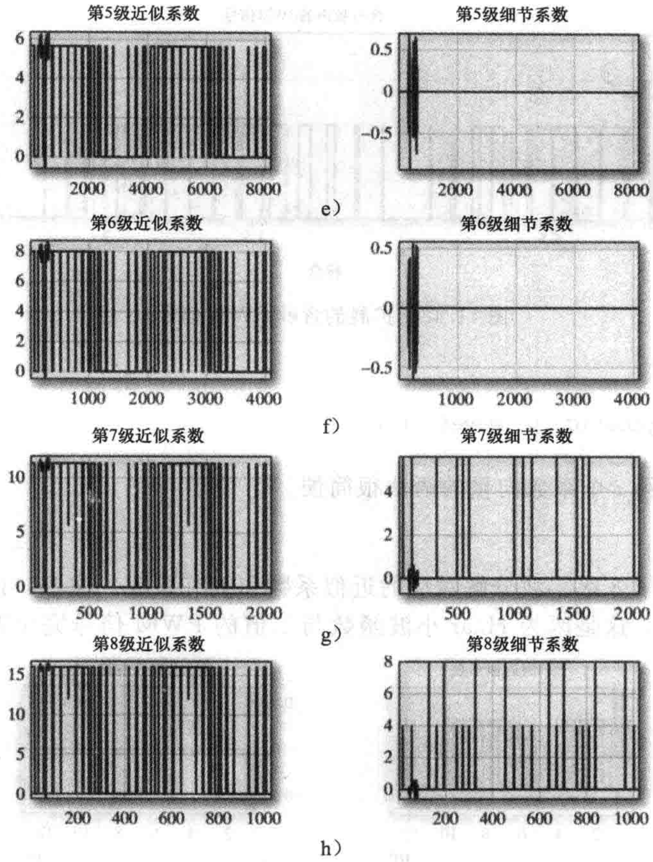


图 11.49 （续）

在进行信号去噪和重建时，可以非常放心地清零这些系数(因为不会造成任何有用信号的损失)。第 7 级和第 8 级细节系数包含某些噪声和信号。我们现在可以对噪声进行阈值化处理，或者我们可以对出现噪声的局部区域直接置零。在近似系数中同样可见噪声的存在，尽管随着级数的增加噪声越来越小。在信号重建时，由于我们仅利用第 8 级近似系数，因此噪声被抑制到最小。

在 MATLAB 中，我们直接置零前 6 级对应的细节系数，而对第 7 级和第 8 级对应的细节系数进行局部阈值化处理。

```
D{1} = D{1}*0;
D{2} = D{2}*0;
D{3} = D{3}*0;
D{4} = D{4}*0;
D{5} = D{5}*0;
D{6} = D{6}*0;
D{7}(75:150) = wthresh(D{7}(75:150), 's', 1);
D{8}(37:75) = wthresh(D{8}(37:75), 's', 1);
```

在通过阈值化修改这些细节系数后，重建信号并绘出最后结果。

```
Cr = [A{s} D{8} D{7} D{6} D{5} D{4} D{3} D{2} D{1}];
fr = waverec(Cr, L, wavef);
plot(fr);
```

图 11.50 显示出最后的重建信号。我们发现噪声并没有被完全消除，但我们去除了大部分噪声。

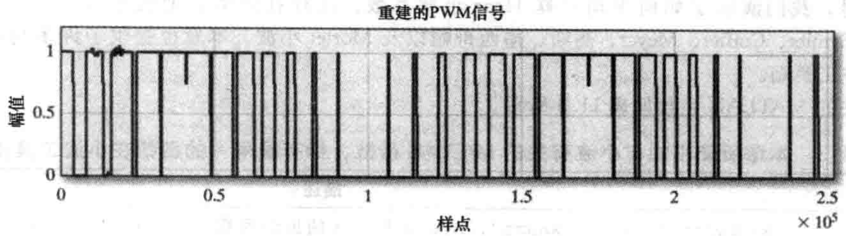


图 11.50 在消除大部分噪声后的 PWM 重建信号

信号压缩。我们发现在利用小波来抑制信号中的噪声时，可以将细节系数中的某些值置为零。我们这样做可以消除噪声，在某些场合，同时也有可能消除部分有用信号。为消除噪声而进行的阈值化处理后，重建的信号包含极少的噪声，但有用信号也存在一些信息损失。可以利用同样的技术来进行信号的有损压缩。

无损压缩是减少对信号编码的位数。当一个无损压缩编码的信号经过传输和相应的解码后，我们可以得到无损的原始信号。而对于有损压缩，我们在信号的信息损失程度与压缩比之间进行平衡。信号的压缩比越高，越有利于信号的传输和存储。有损压缩可以达到10∶1的压缩比(即压缩后的信号是原信号的 1/10)，而无损压缩的压缩比一般是2∶1。

信号的 DWT 将信号分解为近似系数和细节系数。对于某给定信号，用来重建信号的近似系数和细节系数的总点数大致等于信号的样点数。因此，假如我们传输的不是信号本身，而是信号的 DWT 系数，似乎没有任何区别。然而，信号很少是杂乱无序的，信号的能量也常集中在某些频带。该特性使得信号在某些级上的细节系数几乎全部为零。信号能量的集中性使得我们没有必要传输那些已知为零的系数。此外，尽管某些系数不为零，但若其幅值较小，我们可以通过阈值化使之为零，以进一步增加某频带为零系数的数量。对于这些零系数，我们无须直接传输它们的数值，而是传输它们的出现位置。如果这些零系数成系列出现，将需要更少的位数来表示它们。还有其他一些编码和量化技术，其可以进一步降低待传输的信号位数。感兴趣的读者可以参阅 Strang 和 Nguyen 编著的书，以及 Mallat 编著的书，这些书全面介绍了基于小波的信号压缩。

小结

本章简要介绍了小波，并将小波变换与傅里叶变换进行了类比。因为两者都可以看作信号与基信号之间相关的运算。在傅里叶变换中，信号与正弦信号相关，正弦基信号持续时间从负无穷到正无穷，因此在傅里叶变换的结果中，时间变量丢失。在小波变换中，信号与小波相关，而小波基信号的持续时间有限，因而在小波变换的结果中，既有频率变量也有时间变量。为了便于比较小波变换和傅里叶变换，引入了短时傅里叶变换，其可提供该信号的时间信息和频率信息。

小波变换可以是连续的，也可以是离散的。CWT 具有连续的尺度变量，但可以作用于离散的数据，而且可以与计算机的数字计算相兼容。在计算效率方面，CWT 不如 DWT。

对于 DWT 的过程，可以将小波函数当作一个半带的低通或高通滤波器。信号通过低通半带滤波器时，输出的是近似系数。信号通过高通半带滤波器时，输出的是细节系数。这些近似系数可以再通过一组低通和高通半带滤波器，从而得到信号的多维分析。当信号通过一组低通和高通半带滤波器时，我们可以称此种 DWT 为非抽取 DWT。一个可以得到相似结果且计算更加有效的算法就是采用抽取。我们将此种 DWT 称为 DWT。CWT 通过展缩母小波来改变其频率范围以得到连续的尺度(频率数据)。DWT 通过抽取来搬移信号到不同的频带以达到相似的结果。DWT 按 2 的因子抽取，称为二分抽取。

小波具有许多不同的类型，部分设计过程涉及选择恰当的小波来适合其应用对象。Haar 小波是最简

易理解的信号，我们演示了如何手动计算 Haar 小波系数。还存在许多其他的小波系列，本章提及了 Daubechies、Symlet、Coiflet、Meyer、高斯、墨西哥帽以及 Morlet 小波。本章也提供了许多利用 MATLAB 的例题和小波工具箱。

本章相关的 MATLAB 函数如表 11.3 所示。

表 11.3 本章所使用的与小波有关的 MATLAB 函数，标有星号 * 的函数在小波工具箱中

函数	描述
appcoef*	`A= appcoef(C, L, 'wname', N)`；计算 dwt 的近似系数。参数 C 和 L 来自函数 wavedec，'wname'是小波名称，N 是小波分解对应的级数
centfrq*	`FREQ= centfrq('wname', ITER)`；计算小波信号的中心频率。'wname'是小波名称，ITER 是迭代次数，其值为 2 的幂次。返回值是频率，单位为 Hz
cwt*	`coefs= cwt(x, scale, 'wname')`；计算信号 x 的连续小波变换。x 是待分析的信号，scale 是将利用的尺度构成的向量，'wname'是小波名称
detcoef*	`D= detcoef(C, L, N)`；计算 dwt 的细节系数。参数 C 和 L 来自函数 wavedec，N 是小波分解对应的级数
dwt*	`[cA, cD]= dwt(x, 'wname')`；计算一维信号 x 的单级离散小波变换。x 是输入信号，'wname'是小波名称。返回参数 cA 是近似系数，cD 是细节系数。参见 wavedec
dyaddown*	`Y= dyaddown(X, EVENODD)`；计算 x 的 2 倍抽取。若 EVENODD$=0$ 且 $y(k)=x(2k)$. 如 EVENODD$=1$ 则 $y(k)=y(2k+1)$
dyadup*	`y= dyadup(x, EVENODD)`；计算信号 x 的 2 倍内插。如果 EVENODD$=0$，则 $y(2k-1)=x(k)$ 且 $y(2k)=0$。如 EVENODD$=1$，则 $y(2k-1)=0$，且 $y(2k)=x(k)$
gauswavf*	`[PSI, x]= gauwavf(LB, UB, N)`；产生高斯小波函数 ψ。LB 是下限，UB 是上限，N 是样点数量。利用 plot(x，PSI)可显示
imagesc	`imagesc(im)`；将 im 映射到全彩色并显示映射后的图像
mesh	`mesh(x, y, z)`；或 mesh(z)；形成 x、y 和 z 的线框表面图形
mexihat*	`[PSI, X]= mexihat(LB, UB, N)`；产生墨西哥帽小波函数 ψ。LB 是下限，UB 是上限，N 是样点数量。利用 plot(x，PSI)可显示
morlet*	`[PSI, X]= morlet(LB, UB, N)`；产生 Morlet 小波函数 ψ。LB 是下限，UB 是上限，N 是样点数量。利用 plot(x，PSI)可显示
ndwt*	`WT= ndwt(X, N, 'wname')`；产生非抽取的离散小波变换。x 是待变换的信号，N 是变换的级数，'wname'是小波名称。返回的 WT 是一种数据结构(参见 MATLAB 帮助页)
spectrogram	`S= spectrogram(x, window, noverlap, nfft, fs)`；产生短时傅里叶变换。x 是输入信号，window 是窗函数，nfft 是窗长度，noverlap 是相邻段重叠的样点数。nfft 也是每段 FFT 的长度，其值为 2 的幂次。当调用 spectrogram 而没有返回变量时，其显示计算结果的图形
wavedec*	`[C, L]= wavedec(x, N, 'wname')`；计算信号的多级小波分解。x 是输入信号，N 是级数，'wname'是小波名称。C 是分解结果存放的某种数据结构的向量，L 保存向量 C 对应的查询参数
wavefun	`[PHI, PSI, XVAL]= wavefun('wname', ITER)`；返回'wname'指定的小波函数的近似和相应的尺度函数，ITER 为迭代计算的次数。PHI 是尺度函数，PSI 是小波函数，XVAL 是显示尺度函数和小波函数的数值点
waverec*	`x= waverec(C, L, 'wname')`；计算小波重建。利用小波分解产生的向量 C 和 L 重建信号 x
wthresh	`y= wthresh(X, SORH, T)`；对 x 进行阈值化处理。SORH$=$'s'对应软阈值处理，'h'对应硬阈值处理。T 是设定的阈值

习题

概念题

1. 为何小波一般显示“尺度”，而不是频率？
2. 墨西哥帽小波不具有紧支集特性，其含义是什么？

3. 图 P11.3 显示了一个 5 阶多项式，其微分具有 4 个零值点位置。若对该多项式进行 Haar 小波一级变换，这 4 个零斜率点将以怎样的情形出现在小波变换中？
4. 某信号疑含有一个 100Hz 频率的正弦信号，设信号分析的长度不受限制。如果可以利用本章描述的任何小波进行小波变换或者进行傅里叶变换，如何能够最有效地测定该信号中的此正弦信号？说明理由。
5. 图 P11.5 显示了一个频率为 200Hz 正弦信号的连续小波变换，小波函数为墨西哥帽小波。为何此纯正弦信号的 CWT 呈现出一个频带而不是一个单频？

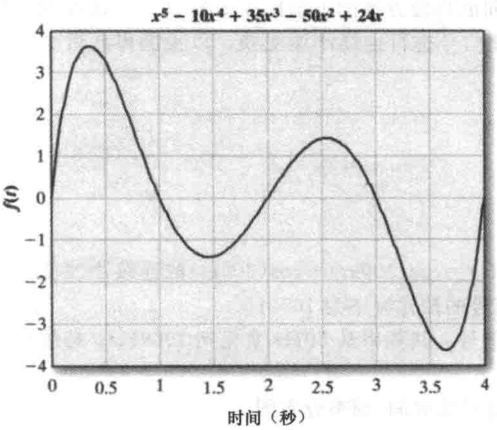


图 P11.3 5 阶多项式

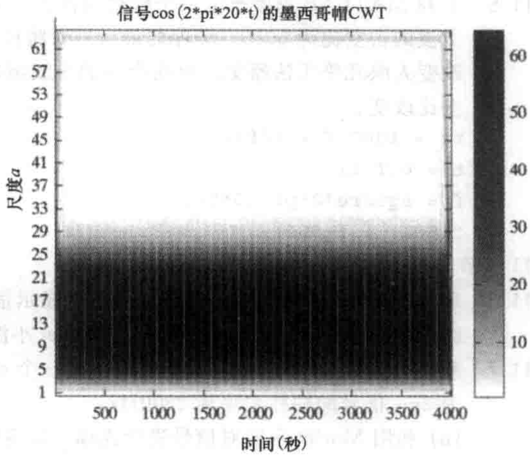


图 P11.5 20Hz 正弦信号的墨西哥帽小波变换

6. 当显示一个正弦信号的连续小波变换时，为何某些小波可得到比其他小波更高的频率分辨率？
7. 传统抽取器由低通滤波器和抽取组成。低通滤波器可以滤除信号中高频分量，以避免抽取后的信号出现频谱混叠。图 11.39(这里的图 P11.7 是其一部分)中的 2 倍抽取前分别有个低通滤波器和高通滤波器，为何没有问题？
8. 在图 11.38 中，假设所用的滤波器为理想的低通和高通半带滤波器。如果这些滤波器是实际的滤波器，在什么情况下可能会遇到混叠问题？
9. 当进行 N 级离散小波变换 DWT 时，根据什么来决定 N 的最大取值？
10. 对于小波变换，为何在低频部分比高频部分具有更高的频率分辨率？

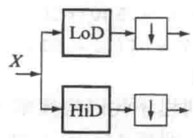


图 P11.7 抽取前的低通滤波器和高通滤波器

11. 关于阈值，
    (a) 相比硬阈值，软阈值有何优点？
    (b) 给出一个信号，说明软阈阈值比硬阈值要好。
    (c) 相比软阈值，硬阈值有何优点？
12. 在采用阈值化处理时，一般将低于设定阈值的数值置为零。反向阈值是将高于设定阈值的数值置为零。给出一个适用于反向阈值化处理的案例。

分析与设计题

11.1 节

11.1 利用 MATLAB 提供的 chirp 函数产生一个信号，其主要参数为 $f_s=2$Hz，起始频率为 0.1Hz，终止频率为 0.8Hz，持续时间为 5s。画出该信号的 STFT 以及 3D 显示。假如窗口的宽度分别为 16、32、64、128 和 256 点。利用 256 点 FFT，其窗口重叠的比例为 50%。解释窗口宽度对时间分辨率和频率分辨率的影响。

11.2 节

11.2 计算给定序列 $f[n]=\{0, 0, 0, 1, 1, -1, 1, 1, 0, 0, 0\}$的一级和二级 Haar 小波变换。利用 MATLAB 提供的函数 cwt 验证你的结果。

11.3 对应于Daub4小波的低通与高通分解滤波器如下

```
LO_D = -0.1294 0.2241 0.8365 0.4830
HI_D = -0.4830 0.8365 -0.2241 -0.1294
```

证明这两个滤波器自身是正交的。

11.4 产生一个持续时间4s的多项式描述的信号 $f(t)=t^3-4t^2-2t+5$，信号的抽样速率为10Hz，利用MATLAB提供的db1、db2、db3和db4小波对其进行连续小波变换。比较和解释不同小波对应的小波变换的图示结果。

11.5 下列MATLAB程序产生一个持续时间为1s、基频为50Hz的方波信号，其抽样频率为1000Hz，方波的占空比为50%。程序的最后一行将位于中间的那段方波的占空比改变为49%，这种微小的改变人眼几乎无法察觉。寻找合适的小波函数对此信号进行连续小波变换，以检测出此微小的占空比改变。

```
fs = 1000;T = 1/fs;
t = 0:T:1;
f = square(2*pi*50*t);
f(501) = f(500);
```

11.3节

11.6 利用MATLAB提供的scal2frq函数，显示信号 $f(t)=\cos(100\pi t)+\cos(300\pi t)$ 的连续小波变换的时间-频率分布图。小波信号选用Morlet小波，信号的抽样频率为1000Hz。

11.7 利用MATLAB提供的chirp函数产生一个chirp信号，其频率从50Hz变化到1000Hz，持续时间为2s，信号的抽样频率为2500Hz。

(a) 利用Morlet小波对信号进行连续小波变换，显示其时间-频率分布图。

(b) 重复(a)，只是chirp信号的频率在2s内从50Hz变化到2000Hz，诠释出现混叠的原因。

11.8 下列程序代码产生一个频率为150Hz正弦信号的2000个样点。然后在样点500～600之间增加一个微小的增量0.01。利用Morlet小波对该信号进行连续小波变换，展现信号中的微小差别。

```
fs = 1000;T = 1/fs;
t = 0:T:2;
f = sin(300*pi*t);
for i = 500:600
f(i) = f(i)+.01;
end
```

11.9 下列程序代码将频率为100Hz的正弦信号分为两部分，两部分之间有一个样点是前一个样点的副本。利用Morlet小波对该信号进行连续小波变换，展现信号中的微小差别。

```
fs = 1000;T = 1/fs;
t = 0:T:2;
f = sin(2*pi*100*t);
f = [f(1:500) f(500) f(501:2000)];
```

11.10 利用Daub4小波演示一个单位脉冲信号通过一级离散小波变换的全过程，并利用MATLAB提供的dwt和idwt函数验证其结果。可以利用MATLAB函数wfilters获得Daub4小波对应的滤波器系数，即[LoD HiD LoR HiR]=wfilters('db2')。分析离散小波变换后的数据，以及重建后的数据。

11.11 分别采用Daub4小波和Haar小波，利用MATLAB对下列定义的信号进行离散小波变换，显示其一级近似系数和细节系数。

```
fs = 10000;T = 1/fs;
t = 0:T:4095*T;
fn = cos(2*pi*6*t);
fn(1250:1255) = fn(1249);
```

该信号是一个频率为6Hz的余弦信号，其在样点1250～1255之间有个毛刺干扰。

11.12 利用MATLAB对习题11.11中的信号进行6级离散小波变换，小波采用Haar小波。并利用Daub4和Symlet 8阶小波函数重做一遍。

11.13 产生一个频率从0Hz变化到1000Hz，持续时间为2s的chirp信号，其抽样频率为2500Hz。基于

图 11.39 所示的滤波器模型，预测该信号对应的细节系数和近似系数可能的结果。利用 MATLAB 和 Daub4 小波对信号进行 6 级离散小波变换，验证你的预测。

11.14 产生一个频率从 100Hz 变化到 3.5kHz，持续时间为 1s 的 chirp 信号，其抽样频率为 8kHz。利用 Morlet 小波对信号进行离散小波变换，计算尺度从 1 至 32 时对应的系数。显示该信号连续小波变换的尺度-时间(或样点)分布图，输出各个尺度与其对应的中心频率转换表。

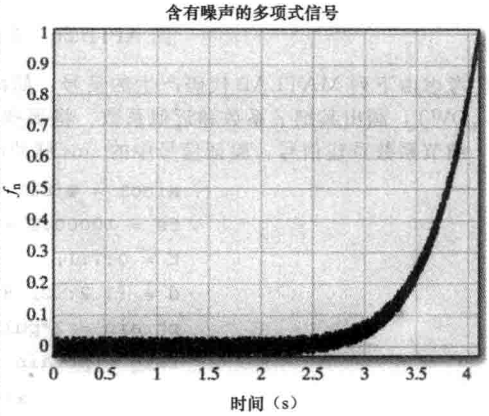
11.15 某信号由下列 MATLAB 代码定义，产生一个如图 11.41 所示的信号分解的尺度-样点分布图，小波采用 db4 小波，分解级数为 8 级。

```
fs = 1000;T = 1/fs;
t = 0:T:4095*T;
f = chirp(t, 10, 4, 400);
```

11.16 下列 MATLAB 代码产生一个信号，其由一个 4 阶多项式表示的信号与随机噪声叠加而成，如图 AP11.16 所示。利用‘Daub16’小波对信号进行 6 级小波分解，计算其近似系数和细节系数。将前 4 级细节系数设置为零，根据修改后的系数重建信号，验证信号中的噪声是否基本被滤除。

```
fs = 1000;T = 1/fs;
t = 0:T:4095*T;
%Create the signal zeros
p1 = [1 1];p2 = [1 .2];
p3 = [1 -1.5];p4 = [1 -3];
x = poly([p1 p2 p3 p4]);
%create the signal
f = polyval(x, t);
mx = max(f);mn = min(f);
f = (f-mn)/(mx-mn);
% %add random noise
a = -.03;b = .03;
fn = f + r = a + (b-a).*rand(1, 4096);
```

a)



b)

图 AP11.16 含有噪声的多项式信号

11.17 对下列信号应用硬阈值和软阈值，阈值设为 0.1。比较阈值化处理前后，信号的最大值、最小值以及均值，并在相同的时间轴上显示该三个信号(原信号、硬阈值后的信号，软阈值后的信号)。

```
t = 0:pi/20:pi;
x = .5*abs(sin(t));
```

11.18 利用连续小波变换，在同一个图中画出 Haar 小波、Daub4 小波、Morlet 小波和高斯小波对应的尺度与频率大致关系。

11.19 利用 Haar 小波和墨西哥帽小波，分别对下列 MATLAB 定义的信号进行连续小波变换。该信号由一系列高度和宽度不等的水平线组成。观察此信号的连续小波变换，你能从中推断出什么？

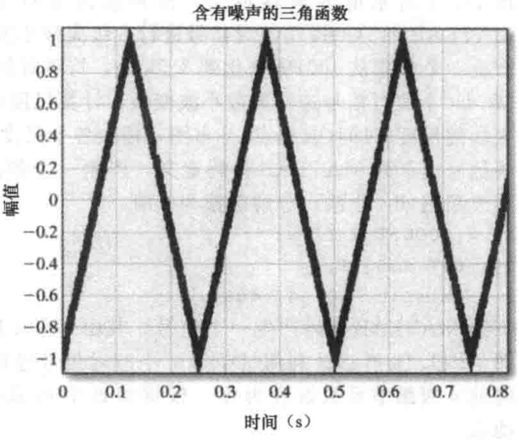
```
hr = [50 80 100 75 60 125 40 30];
wr = [80 60 80 100 50 50 100 60];
f = hr(1)*ones(1, wr(1));
for i=2:length(hr);
    f = [f hr(i)*ones(1, wr(i))];
end
```

11.4 节

11.20 下列 MATLAB 代码产生一个叠加了噪声的三角波信号(见图 AP11.20)，利用 Daub4 小波对信号进行 6 级离散小波变换。对 6 级细节系数进行软阈值化处理，1 级和 2 级的阈值设置为 0.3，3 级和 4 级的阈值设置为 0.2，5 级和 6 级的阈值设置为 0.1，随着每级的阈值化处理，分别重建相应的信号，观察 6 个不同的重建信号的效果。

```
fs = 10000;T = 1/fs;
t = 0:T:8191*T;
fsig = sawtooth(2*pi*4*t, .5);
a = -.1;b = .1;
r = (a + (b-a).*rand(1, 8192));
fsig = fsig + r;
```

a)



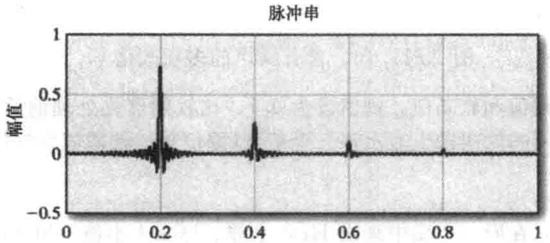
b)

图 AP11.20 含有噪声的三角波

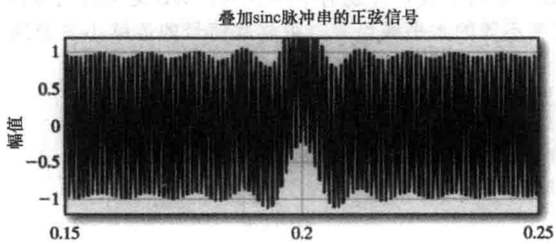
11.21 考虑由下列 MATLAB 代码产生的信号，如图 AP11.21 所示。利用'sym8'小波对信号进行 6 级 DWT，画出其细节系数和近似系数。将那些几乎不包含正弦分量的系数置为零，根据修改后的细节系数重建信号，验证信号中的 sinc 脉冲串已基本被滤除。

```
sinc1 = @(t1) sinc(200*t1);
fs = 10000;T = 1/fs;
t = 0:T:1;
d = [(.2:.2:.8); exp(-1:-1:-4)]';
ptrain = 2*pulstran(t, d, sinc1);
fsig = ptrain + sin(2*pi*1000*t);
```

a) 样点



b) 样点



c) 样点

图 AP11.21 sinc 脉冲串作为噪声叠加于正弦信号

11.22 假设我们有个样点数为 8192 的信号，每个样点是一个 4 字节的浮点数。

(a) 需要多少存储单元(以字节为单位)来存储该信号？

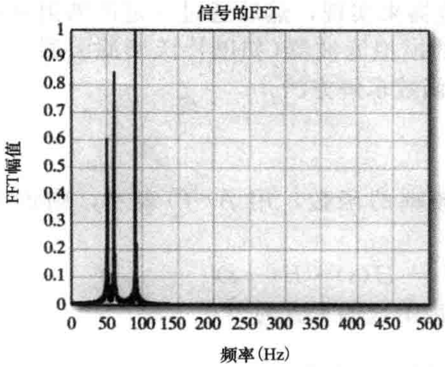
(b) 假如利用 Daub4 小波对该信号进行 8 级 DWT，需要多少存储单元来存储此 8 级的细节系数

和最后一级的近似系数？

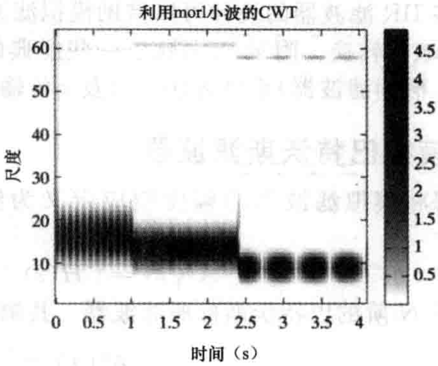
(c) 如果第 5 级、第 6 级、第 7 级和第 8 级的细节系数基本为零，需要多少存储单元来存储前 4 级的细节系数和最后一级的近似系数？

11.23 对于给定的信号 $f(t)=t^5-10.4t^4+39.68t^3-68.08t^2+51.48t-13.68$，利用 Haar 小波对该信号进行至少 64 级的连续小波变换，测定该信号在 $0\leqslant t\leqslant 4$ 区间上何处的微分值等于零。

11.24 此处有两个图，图 AP11.24a 是信号对应的 FFT，图 AP11.24b 是利用 Morlet 小波对信号进行 64 级连续小波变换的结果，根据这两幅图，关于该信号你能得到什么结论？



a）FFT



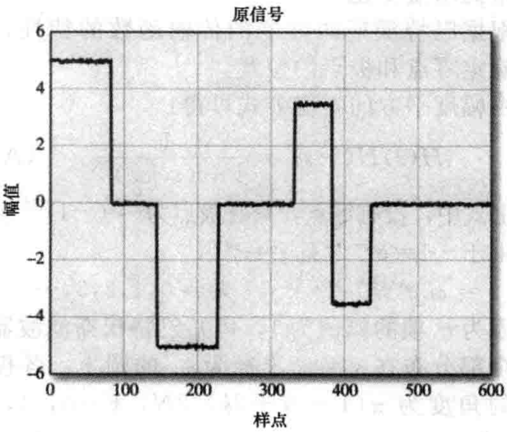
b）小波变换

图 AP11.24 信号的 FFT 和连续小波变换

11.25 图 AP11.25 所示为含有随机噪声的信号，利用 Haar 小波对该信号进行两级离散小波变换，然后对细节系数进行软阈值化处理，阈值设置为 0.2，显示去噪后的信号。

```
hr = [5 0 -5 0 3.5 -3.5 0 0];
wr = [80 60 80 100 50 50 100 60];
f = hr(1)*ones(1, wr(1));
for i=2:length(hr);
f = [f hr(i)*ones(1, wr(i))];
end
f = f + .1*rand(1, length(f)) - .1;
```

a）



b）

图 AP11.25 通过软阈值滤除信号中的加性噪声

11.26 试证明 Haar 小波是唯一满足对称性和正交性的小波。若具有对称性，其滤波器系数应为下列格式

$$\{b_0,\ b_1,\ b_2,\ \cdots,\ b_{L/2-1},\ b_{L/2-1},\ \cdots,\ b_2,\ b_1,\ b_0\}$$

其中，L 为滤波器的长度。若具有正交性，所有按 2 的倍数平移的内积都为零。为保证长度 L 不为零，系数 b_0 不能为零。为了证明 Haar 小波是唯一满足这些约束条件的小波，构成该系数与平移 $L-2$ 的系数之内积，再按 $L-4$ 平移等重复此过程，最终将得到 $b_1=b_2=\cdots=b_{L/2-1}=0$。

附录 A 模拟滤波器设计

许多 IIR 滤波器的设计可以借助模拟滤波器来实现，然后通过一定的映射函数完成从 s 域到 z 域的转换。附录 A 回顾了一些经典的模拟滤波器(如巴特沃斯滤波器、切比雪夫滤波器、椭圆滤波器)设计方法，以及 s 传输函数求解方法。

A.1 模拟巴特沃斯滤波器

如果将模拟滤波器的幅度响应定义为频率的函数，用 $A(f)$表示，则该响应的平方为[1]

$$A^2(f)=|H(s)|^2=H(s)\cdot H(-s) \tag{A.1}$$

对于 N 阶的巴特沃斯低通滤波器，其幅度响应的平方为[1]

$$A^2(f)=\frac{1}{1+(f/f_c)^{2N}} \tag{A.2}$$

式中，N 是滤波器的阶数，f_c 表示 3dB 处截止频率。该幅度特性的特点是其通频带最大平坦，而且在所有频带内是单调变化的。图 A.1 给出了一种典型的巴特沃斯滤波器的幅频响应特性。对于该巴特沃斯滤波器，在通频带内，频率响应曲线没有起伏，通带波动定义为通带的增益变化。

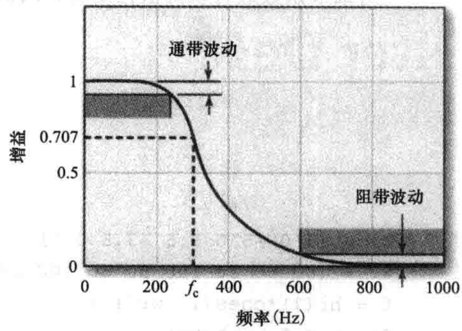


图 A.1 四阶模拟巴特沃斯滤波器(截止频率 300Hz)的幅频响应

根据巴特沃斯滤波器的传输函数的特性，很容易确定零点和极点的位置。

由幅度平方的响应公式可得：

$$H(s)H(-s)=\frac{1}{1+(s/j\omega_c)^{2N}} \tag{A.3}$$

上式中，没有零点，只有极点为$s=(-1)^{1/2N}j\omega_c$。由于$-1=e^{j\pi+2k\pi}$且 $j=e^{j\pi/2}$，

$$s=\omega_c e^{j\pi(1+N+2k)/(2N)},\quad k=0,1,2,3,\cdots$$

因为 e^j 项的幅值为 1，可知巴特沃斯滤波器的极点全部分布在 s 平面半径为ω_c 的圆上。各极点对应的角度为 $\pi(1+N+2k)/2N$，$k=0$，1，⋯ $(2N-1)$。

在求解极点位置时，使用了幅度响应的平方，因此所求的极点数是相应幅度函数极点数的两倍。由于幅度响应平方等于 $H(s)$与 $H(-s)$的乘积，将左半平面极点属于$H(s)$，右半平面的极点属于 $H(-s)$，则可求得 $H(s)$的极点位置为

$$s_k=\omega_c e^{j\pi(1+N+2k)/(2N)},k=0,1,\cdots,N-1 \tag{A.4}$$

例如，一个截止频率为 55Hz 的四阶巴特沃斯滤波器的极点如图 A.2 和表 A.1 所示，其传输函数为

$$H(s)=\frac{1}{s^4+903.03s^3+4.0773\times10^5s^2+1.0784\times10^8s+1.4262\times10^{10}}$$

截止频率为 55Hz 的四阶巴特沃斯低通滤波器的幅频响应如图 A.3 所示。

表 A.1 截止频率为 55Hz 的四阶巴特沃斯低通滤波器的极点位置

极点	R	θ	实部	虚部
1	110π	5π/8	−132.246	j319.270
2	110π	7π/8	−319.270	j132.246
3	110π	9π/8	−319.270	−j132.246
4	110π	11π/8	−132.246	−j319.270

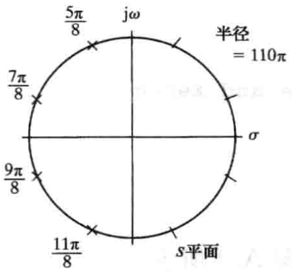


图 A.2 截止频率为 55Hz 的四阶巴特沃斯滤波器 s 平面极点分布图

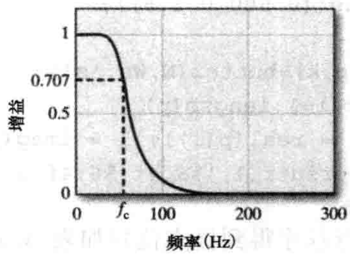


图 A.3 截止频率为 55Hz 的四阶巴特沃斯低通滤波器的幅频响应

此例告诉我们如何设计一个阶数已知的滤波器，但通常情况下，所给的滤波器参数并不包含阶数，而只给出滤波器通带和阻带的边界以及它们的波动，由这些已知条件和巴特沃斯滤波器的幅度响应(式(A.2))，可直接求出滤波器的阶数。

为了根据频率指标计算滤波器的阶数，定义以下符号。

f_{pass}：通带边界频率(常称通带截止频率)(Hz)

f_{stop}：阻带边界频率(常称阻带截止频率)(Hz)

R_{pass}：通带波动

R_{stop}：阻带波动

根据以上定义，可知阻带截止频率 f_{stop} 的幅度响应为 R_{stop}，通带截止频率 f_{pass} 处的幅度响应为 $1-R_{pass}$，结合式(A.2)可得：

$$R_{stop}^2 = \frac{1}{1+(f_{stop}/f_c)^{2N}} \tag{A.5}$$

$$(1-R_{pass})^2 = \frac{1}{1+(f_{pass}/f_c)^{2N}} \tag{A.6}$$

由上面两个方程可以求解得到 N 和 f_c：

$$N = \frac{\ln(K_s/K_p)}{2\ln(f_{stop}/f_{pass})} \tag{A.7}$$

$$\ln(f_c) = \frac{\ln(K_s)\ln(f_{pass}) - \ln(K_p)\ln(f_{stop})}{\ln(K_s/K_p)} \tag{A.8}$$

式中

$$K_s = (1/R_{stop}^2) - 1, \quad K_p = \frac{1}{(1-R_{pass})^2} - 1$$

上述滤波器的设计过程可以利用 MATLAB 实现。下面的 MATLAB 程序为巴特沃斯低通滤波器设计，该滤波器的参数为 $R_s=0.01$，$R_p=0.01$，$f_p=150$，$f_s=200$。

```
% butter.m
Rs = .01; Rp = .01; % Stop and pass band ripple
Rsdb = -20*log10(Rs);Rpdb = -20*log10(1-Rp); % db ripple
```

```
fp = 150; fs = 200;Wp = 2*pi*fp;Ws = 2*pi*fs;
% buttord finds the order for a Butterworth filter.
[N, Wn] = buttord(Wp, Ws, Rpdb, Rsdb, 's');
[num,den]=butter(N,Wn,'s'); % butter finds the filter
f = linspace(0,500);
w = 2*pi*f;
h=freqs(num,den,w);
plot(f,abs(h),'k')
axis([0 500 0 1.5]);
%
[z,p,k]=butter(N,Wn,'s');    % print the poles and zeros
for i=1:length(p)
  R = real(p(i)); I = imag(p(i));
  fprintf(1,'%6.4f %6.4f\n', R, I);
end
```

运行程序得到极点位置如表 A.2 所示，幅频响应如图 A.4 所示。

表 A.2 满足 $R_s=0.01$，$R_p=0.01$，$f_p=150$，$f_s=200$ 指标的巴特沃斯低通滤波器的极点位置

实部	虚部	实部	虚部
−70.0579	±j1024.2101	−877.1498	±j533.4066
−208.8686	±j1005.1310	−796.3478	±j647.8770
−343.7885	±j967.3282	−700.7114	±j750.2787
−1017.0415	±j139.7891	−592.0220	±j838.7040
−988.5341	±j276.9743	−472.3043	±j911.5060
−941.6122	±j408.9999	−1026.6034	0.0000

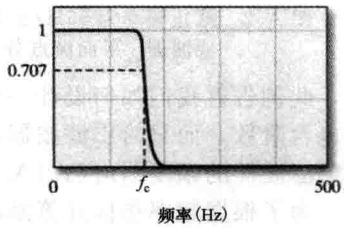


图 A.4 表 A.2 所列极点对应的巴特沃斯低通滤波器的幅频响应图

A.2 模拟切比雪夫滤波器

低通切比雪夫滤波器也是全极点滤波器，在很多方面与巴特沃斯滤波器很相似。两者之间最主要的区别在于它们的通带和过渡带。巴特沃斯滤波器在通频带中有最大平坦的幅度特性，但过渡带相对较宽。切比雪夫滤波器在通带内有一定波动，且过渡带较窄。在大多数情况下，如果不要求通带具有很大的平坦度，使用切比雪夫方法设计的滤波器阶数较低，因而计算效率更高。

切比雪夫滤波器的幅度响应平方函数为

$$A^2(f)=\frac{1}{1+\varepsilon^2 V_N^2(f/f_{\text{pass}})} \tag{A.9}$$

式中，ε 是与通带波动有关的参量，f_{pass} 是通带截止频率，$V_N(x)$ 是切比雪夫多项式，定义为

$$V_N(x)=\cos(N\arccos(x))$$

切比雪夫可以通过以下关系式展开：

$$V_0(x)=1$$
$$V_1(x)=x$$
$$V_{k+1}(x)=2xV_k(x)-V_{k-1}(x)$$

由上述等式，可以递推出切比雪夫多项式

$$V_0(x)=1$$
$$V_1(x)=x$$

$$V_2(x) = 2x^2 - 1$$
$$V_3(x) = 4x^3 - 3x$$
$$V_4(x) = 8x^4 - 8x^2 + 1$$
$$V_5(x) = 16x^5 - 20x^3 + 5x$$
$$\vdots$$

ε 是与通带波动有关的参量，它与 R_p 的关系如下

$$R_p = 1 - \frac{1}{\sqrt{1+\varepsilon^2}}$$

因此根据通带波动，可以求得 ε 为

$$\varepsilon = \sqrt{\frac{1}{(1-R_p)^2} - 1} \tag{A.10}$$

可以证明切比雪夫滤波器的极点分布在 s 平面的一个椭圆上[1,3]。这个椭圆由两个圆构成，如图 A.5 所示。椭圆的长轴和短轴分别为

$$r_{\text{major}} = 2\pi f_{\text{pass}} \frac{C^2+1}{2C} \tag{A.11}$$

$$r_{\text{minor}} = 2\pi f_{\text{pass}} \frac{C^2-1}{2C} \tag{A.12}$$

式中

$$C = \left(\frac{1}{\varepsilon} + \sqrt{1+1/\varepsilon^2}\right)^{1/N} \tag{A.13}$$

各极点的角度与巴特沃斯滤波器极点的角度一样，等于 $\pi(1+N+2k)/2N$，$k=0, 1, \cdots, (2N-1)$。四阶切比雪夫滤波器的极点分布如图 A.5 所示。

分析图 A.5，可以利用几何知识写出极点的形式：

$$\theta_k = \pi(1+N+2k)/2N, \quad k = 0,1,\cdots,(2N-1)$$
$$x_k = r_{\text{minor}} \cos(\theta_k)$$
$$y_k = r_{\text{major}} \sin(\theta_k)$$

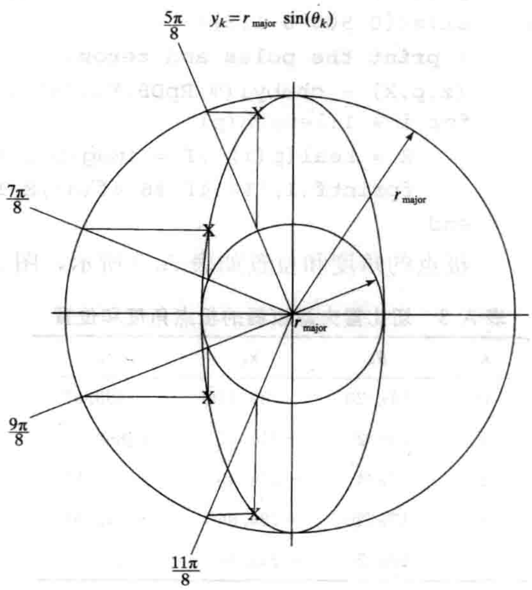


图 A.5 四阶切比雪夫滤波器的极点位于 s 平面的椭圆上，该椭圆由两个半径分别为 r_{major} 和 r_{minor} 的圆构成

为了得到满足条件的切比雪夫滤波器的最小阶数，需求解方程(A.9)得到 N。不同于巴特沃斯滤波器，切比雪夫滤波器的幅度平方方程由通带截止频率表示，而不是 3dB 处的截止频率。我们通过阻带截止频率求解方程(A.9)中的 N，阻带截止 f_{stop} 处的通带波动为 R_{stop}，代入方程(A.9)可得

$$R_{\text{stop}}^2 = \frac{1}{1+\varepsilon^2 V_N^2(f_{\text{stop}}/f_{\text{pass}})}$$

根据上式求解 $V_N{}^2$

$$V_N^2(f_{\text{stop}}/f_{\text{pass}}) = \frac{1-R_{\text{stop}}^2}{R_{\text{stop}}^2 \varepsilon^2} \tag{A.14}$$

切比雪夫多项式定义为

$$V_N(x) = \cos(N\arccos(x))$$

此时，x 等于 $f_{\text{stop}}/f_{\text{pass}}$，为大于 1 的正数。因此，上式可写为

$$V_N(x) = \cosh(N\text{arccosh}(x))$$

式中，$\cosh(x)$是双曲余弦函数，定义为

$$\cosh(x)=\frac{e^{x}+e^{-x}}{2}=\cos(jx)$$

将上式代入方程(A.14)，解得 N 为

$$N=\frac{\operatorname{arccosh}\left[\frac{(1-R_{stop}^{2})^{1/2}}{R_{stop}\varepsilon}\right]}{\operatorname{arccosh}(f_{stop}/f_{pass})}\tag{A.15}$$

上式即可求出最小阶数 N，若该值不为整数时，N 值通常取大于它的最小整数。

在设计巴特沃斯滤波器时，可以利用 MATLAB 编程实现。同样也可以利用 MATLAB 设计切比雪夫滤波器。下面的 m 文件给出了相应的程序，实现 $f_{pass}=150$Hz，$f_{stop}=200$Hz，$R_{pass}=0.01$，$R_{stop}=0.01$ 的切比雪夫滤波器的设计。

```
% Cheby.m
fstop = 200; fpass = 150; %Initialize design parameters
Rs = .01; Rp = .01;
RpDB = -20*log10(1-Rp);    % change ripple to decibels
RsDB = -20*log10(Rs);
Wp = fpass*2*pi;Ws = fstop*2*pi;
% cheblord calculates the order for a Chebyshev filter.
[N Wn] = cheb1ord(Wp, Ws, RpDB, RsDB,'s');
[num,den]= cheby1(N,RpDB,Wp,'s'); %cheby1 finds the filter.
f = linspace(0,500);
w = 2*pi*f;
h = freqs(num,den,w); % freqs finds the magnitude vector h
plot(f,abs(h),'k')       % at frequencies w
axis([0 500 0 1.5]);
% print the poles and zeros.
[z,p,k] = cheby1(N,RpDB,Wp,'s');
for i = 1:length(p)
    R = real(p(i));I = imag(p(i));
    fprintf(1,'%6.4f %6.4f\n',R,I);
end
```

极点的角度和位置如表 A.3 所示，图 A.6 为该滤波器的幅度响应。

表 A.3　切比雪夫滤波器的极点角度和位置

k	θ_k	x_k	y_k
0	$11\pi/20$	-39.478	$\pm j963.67$
1	$13\pi/20$	-114.57	$\pm j869.34$
2	$3\pi/4$	-178.45	$\pm j689.91$
3	$17\pi/20$	-224.86	$\pm j442.95$
4	$19\pi/20$	-249.26	$\pm j152.63$

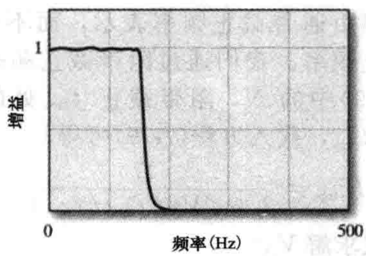


图 A.6　$N=10$ 时切比雪夫低通滤波器的幅频响应曲线

现在来比较巴特沃斯滤波器与切比雪夫滤波器的设计过程。设计巴特沃斯滤波器时，截止频率 f_c 是很重要的参数。对于一个巴特沃斯低通滤波器，截止频率通常定义为幅度增益下降到 0.707(即 3dB)时所对应的频率值，因此该截止频率称为 3dB 截止频率。而对于一个切比雪夫滤波器，较为重要的参数是通带截止频率 f_{pass}。若将通带波动定义为 R_{pass}，指定 0Hz 处的幅度为 1，则 f_{pass}处的增益为 $1-R_{pass}$。因此，如果同时给出通带和阻

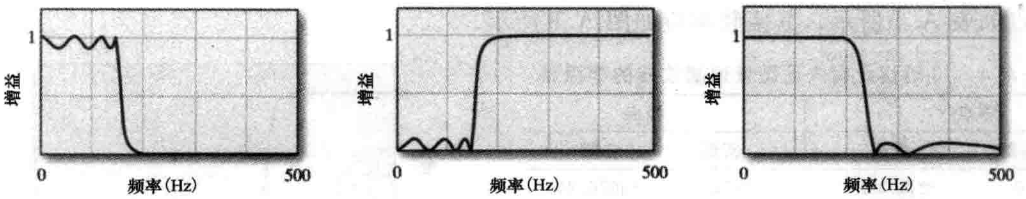
带的截止频率以及它们的波动，设计巴特沃斯滤波器比切比雪夫滤波器多一个步骤，即求出截止频率。

设计巴特沃斯滤波器和切比雪夫滤波器的差别还体现在最终的结果中。在前面的例子中，为了满足相同的滤波器指标要求，切比雪夫滤波器在通带具有明显的波纹，且滤波器的最小阶数为 10。而对于巴特沃斯滤波器，最小阶数为 23，该结果具有一般性。在大多数情况下，如果不要求通带具有很大的平坦度，可以用较低阶的切比雪夫滤波器实现。

A.3　模拟反切比雪夫滤波器

可以用完全相反的方式构建一个切比雪夫滤波器。在这种情况下，滤波器的通带和阻带特性互换。滤波器的幅度特性在通带内有波动，在阻带内单调下降，称为切比雪夫Ⅰ型滤波器；在通带内单调下降，而在阻带内有波动，则称为切比雪夫Ⅱ型滤波器，即反切比雪夫滤波器。

切比雪夫Ⅱ型滤波器是切比雪夫Ⅰ型滤波器的拓展。图 A.7a 是一个四阶的切比雪夫Ⅰ型滤波器的幅频响应图。在通带内有明显的波纹效果，在阻带内单调地变化。若用常数 1 减去该幅度响应，则得到如图 A.7b 所示的响应曲线。此时，在通带内是单调的，在阻带内是波动的，但滤波器由原来的低通滤波器变为高通滤波器。为了得到一个切比雪夫Ⅱ型低通滤波器，需计算高频和低频的频率映射关系，可在原方程中用 $1/f$ 替换 f 实现。所得结果即为所需的幅度响应，如图 A.7c 所示。



a）四阶切比雪夫Ⅰ型滤波器的幅频响应　b）用1减去切比雪夫Ⅰ型滤波器幅值所得幅频响应　c）将低频和高频重新映射后所得切比雪夫Ⅱ型滤波器的幅频响应

图 A.7　切比雪夫Ⅱ型滤波器的推导

式(A.9)给出的切比雪夫滤波器的幅度平方响应为

$$A^2(f)=\frac{1}{1+\varepsilon^2V_N^2(f/f_{\text{pass}})}$$

用 1 减去上式得

$$1-A^2(f)=\frac{\varepsilon^2V_N^2(f/f_{\text{pass}})}{1+\varepsilon^2V_N^2(f/f_{\text{pass}})}$$

将频率变量倒置得到切比雪夫Ⅱ型低通滤波器的幅度平方响应为

$$A^2(f)=\frac{\varepsilon_{\text{inv}}^2V_N^2(f_{\text{stop}}/f)}{1+\varepsilon_{\text{inv}}^2V_N^2(f_{\text{stop}}/f)} \tag{A.16}$$

由于频率的倒置，将 f_{pass} 替换为 f_{stop}。

为确定 ε_{inv} 的值，令 $A^2(f_{\text{stop}})=\frac{1}{1+\varepsilon^2}$，该等式要求

$$\varepsilon_{\text{inv}}=\frac{1}{\varepsilon V_N(f_{\text{stop}}/f_{\text{pass}})} \tag{A.17}$$

类似地，可以定义一个新的频率变量 η，用它来替换倒置的频率 f，即

$$\eta=\frac{\omega_{\text{stop}}\omega_{\text{pass}}}{\omega} \tag{A.18}$$

将上式代入式(A.16)得

$$A^2(f)=\frac{\varepsilon_{\text{inv}}^2 V_N^2(\eta/\omega_{\text{pass}})}{1+\varepsilon_{\text{inv}}^2 V_N^2(\eta/\omega_{\text{pass}})} \tag{A.19}$$

式(A.19)中的分母与切比雪夫Ⅰ型滤波器的分母形式一致。因此可以用相同的方法求解极点，这些极点是频率变量η的极点，需要通过方程(A.18)重新映射。

切比雪夫Ⅱ型滤波器的系统函数有零点，零点为下述方程的根，

$$\varepsilon_{\text{inv}}^2 V_N^2(\eta/\omega_{\text{pass}})=0$$

根据切比雪夫多项式的定义可得

$$V_N(\eta/\omega_{\text{pass}})=\cos(N\arccos(\eta/\omega_{\text{pass}}))=0$$

或

$$N\arccos(\eta/\omega_{\text{pass}})=(2k+1)(\pi/2)$$

从而得到

$$\eta/\omega_{\text{pass}}=\cos\left(\frac{(2k+1)\pi}{2N}\right)=\frac{\omega_{\text{stop}}\cdot\omega_{\text{pass}}}{\omega_k\cdot\omega_{\text{pass}}}$$

在s平面上，零点的位置为

$$\omega_k=\frac{\omega_{\text{stop}}}{\cos\left[\frac{(2k+1)\pi}{2N}\right]},\quad k=0,1,2,\cdots,N-1 \tag{A.20}$$

上述切比雪夫Ⅱ型滤波器的设计过程可以用MATLAB实现。以下m文件中的程序设计满足：$f_{\text{pass}}=150\text{Hz}$，$f_{\text{stop}}=200\text{Hz}$，$R_{\text{pass}}=0.01$，$R_{\text{stop}}=0.01$指标的滤波器。相应的零极点如表A.4所示，其幅频响应如图A.8所示。

表 A.4　10阶切比雪夫Ⅱ型低通滤波器的零极点

零点		极点	
实部	虚部	实部	虚部
0	±j8032.0	−2071.8	±j676.24
0	±j1272.3	−1208.69	±j1269.2
0	±j2767.0	−610.32	±j1257.7
0	±j1777.2	−287.34	±j1162.1
0	±j1410.4	−84.998	±j1105.9

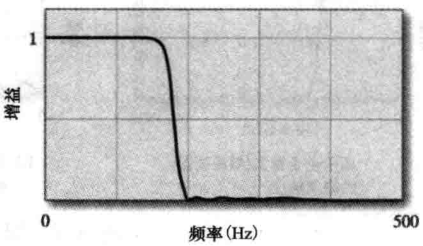


图 A.8　利用MATLAB设计10阶切比雪夫Ⅱ型低通滤波器的幅频响应

注意，在设计切比雪夫Ⅱ型滤波器时，需要明确给出阻带波动和阻带截止频率。MATLAB产生的切比雪夫Ⅱ型滤波器的幅频特性曲线经过(f_{stop}，R_{S})点，若与MATLAB程序相一致，要令式(A.19)中的$\varepsilon_{\text{inv}}=R_{\text{S}}^2/(1-R_{\text{S}}^2)$。

```
% ChebyINV.m
fstop = 200;fpass = 150; % Initialize the specifications.
Rs = .01;Rp = .01;
Ws = fstop*2*pi;Wp = 2*pi*fpass;
RsDB = -20*log10(Rs); % Stop band ripple in decibels
RpDB = -20*log10(1-Rp);
% cheb2ord finds the order
[N Wn] = cheb2ord(Wp,Ws,RpDB,RsDB,'s');
% cheby2 finds the inverse Chebyschev filter.
[num,den]=cheby2(N,RsDB,Ws,'s');
f = linspace(0,500); % f is the frequency vector in Hertz
w = 2*pi*f;
h=freqs(num,den,w); % freqs finds the magnitude coeff w.
```

```
plot(f,abs(h),'k')  % plot h vs f in color black = 'k'
axis([0 500 0 1.5]);
% print the poles and zeros.
[z,p,k] = cheby2(N,RsDB,ws,'s');
for i = 1:length(z)
    R = real(z(i));I = imag(z(i));
    fprintf(1,'%6.4f %6.4f\n',R,I);
end
for i = 1:length(p)
    R = real(p(i));I = imag(p(i));
    fprintf(1,'%6.4f %6.4f\n',R,I);
end
```

A.4 模拟椭圆滤波器

巴特沃斯和切比雪夫模拟低通滤波器都是“全极点”滤波器(切比雪夫Ⅱ型滤波器除外)。它们的阻带都是单调的。在通带内，巴特沃斯滤波器的极点分布使得其具有最大平坦特性，而切比雪夫滤波器在通带内允许有等波纹波动，其极点的分布使得过渡带比巴特沃斯滤波器更陡峭。椭圆滤波器进一步拓展了切比雪夫滤波器的思想，同时允许通带和阻带内都具有等波纹波动，因此，其同时具有零点和极点，其零极点的位置很近，使得设计相同阶数的滤波器时，椭圆滤波器的过渡带最陡峭。

椭圆滤波器由 Wilherm Cauer 提出，最早用于德国电话行业(椭圆滤波器又称为 Cauer 滤波器)。之所以称为椭圆滤波器是因为它依赖于第一类椭圆积分，即[4]

$$f(\varphi,k)=\int_0^{\varphi}\frac{\mathrm{d}\theta}{\sqrt{(1-k\cdot\sin^2(\theta))}}$$

椭圆积分不能以闭合的形式进行积分，可以写为椭圆函数形式(称为雅克比椭圆函数)，但可将积分表示为无限个序列。因此，椭圆滤波器的设计借助于截断无穷序列的近似方法或查找表格实现。

椭圆滤波器在通带和阻带内均具有等波纹特性，“等波纹”是指频带中的波纹的振幅是均匀的。在设计 FIR 滤波器时，我们知道滤波器的这种等波纹特性在某种意义上是最佳的。可以证明，设计满足给定指标的滤波器，用椭圆滤波器实现的滤波器阶数最低。

椭圆滤波器的难点在于其设计，它的推导和设计过程涉及非常复杂的数学计算，本书在这里不做详细介绍。它严谨的推导过程见参考文献[3]和[5]，参考文献[3]描述了用近似椭圆积分值的方法得到滤波器传输函数的设计过程。

利用 MATLAB 设计椭圆滤波器，其满足以下指标：$f_{pass}=150\text{Hz}$，$f_{stop}=200\text{Hz}$，$R_{pass}=0.01$，$R_{stop}=0.01$，其程序如下所示。注意，这些指标与前面介绍的巴特沃斯滤波器、切比雪夫滤波器、反切比雪夫滤波器的设计指标是相同的。由 m 文件运行得到的零极点列在表 A.5 中，滤波器的幅频特性如图 A.9 所示。

表 A.5 三阶低通椭圆模拟滤波器的零极点 $f_{pass}=150\text{Hz}$，$f_{stop}=200\text{Hz}$，$R_{pass}=0.01$，$R_{stop}=0.01$

零点		极点	
实部	虚部	实部	虚部
0	±j1344.6	−609.06	±j467.08
0	±j3989.8	−265.50	±j989.72
0	±j1650.1	−62.32	±j1129.3

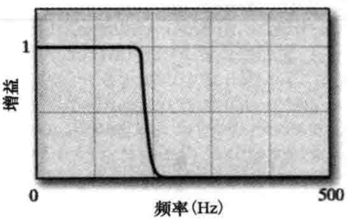


图 A.9 低通椭圆滤波器的幅频响应图，其零极点如表 A.5 所示

三阶椭圆滤波器可以满足上述指标要求，而要满足同样要求，巴特沃斯滤波器是23阶，切比雪夫滤波器是10阶。

```
% Elliptic.m
fs = 200;fp = 150; % Initialize specification variables.
wp = 2*pi*fp; ws = 2*pi*fs;
Rs = .01;Rp = .01;
RpDB = -20*log10(1-Rp);RsDB = -20*log10(Rs);
wc = (wp*ws)^.5; % wc is the cutoff frequency.
%
N = ellipord(wp,ws,RpDB,RsDB,'s'); % Find the order.
[num,den] = ellip(N,RpDB,RsDB,wc,'s'); % Finds the coeff.
f = linspace(0,500,1000); % create a frequency vector
w = 2*pi*f;
h=freqs(num,den,w); % freqs finds the magnitude coeff
plot(f,abs(h),'k') % draw the plot
axis([0 500 0 1.5]);
% Find and print the poles and zeros
[z,p,k] = ellip(N,RpDB,RsDB,wc,'s');
for i = 1:length(z)
    R = real(z(i));I = imag(z(i));
    fprintf(1,'%6.4f %6.4f\n',R,I);
end
for i = 1:length(p)
    R = real(p(i));I = imag(p(i));
    fprintf(1,'%6.4f %6.4f\n',R,I);
end
```

A.5 模拟滤波器特性总结

模拟滤波器的种类繁多，其中巴特沃斯滤波器、切比雪夫Ⅰ型滤波器、切比雪夫Ⅱ型滤波器、椭圆滤波器最为常用。这四种滤波器的特性如表A.6所示。

表A.6 四种典型滤波器的特性

滤波器类型	通带	过渡带	阻带	评价
巴特沃斯	最大平坦	宽	单调	全极点滤波器，设计过程简单，阶数较高才满足一定指标，阻带有多余衰减
切比雪夫	等波纹	中等	单调	全极点滤波器，设计过程简单，阶数高于椭圆滤波器，低于巴特沃斯滤波器，阻带有多余衰减
逆切比雪夫	单调	中等	等波纹	设计过程比切比雪夫Ⅰ型滤波器稍复杂，阶数与切比雪夫Ⅰ型滤波器相同
椭圆滤波器	等波纹	窄	等波纹	数学推导复杂，相同指标时椭圆滤波器的阶数比巴特沃斯和切比雪夫滤波器低很多

附录B

参考文献

第1章

1. F. M. Smits, ed., *A History of Engineering and Science in the Bell System—Electronics Technology (1925–1975)*, New York: AT&T Bell Laboratories (Author), 1985.

第2章

1. C. L. Phillips, J. M. Parr, and E. A. Riskin, *Signals, Systems, and Transforms*, 4th ed., Upper Saddle River, NJ: Pearson Prentice Hall, 2008.
2. C. L. Phillips and H. T. Nagle, *Digital Control System Analysis and Design*, 3rd ed., Upper Saddle River, NJ: Pearson Prentice Hall, 1996.

第3章

1. R. A. Gabel and R. A. Roberts, *Signals and Linear Systems*, New York: Wiley, 1987.
2. L. B. Jackson, *Signals, Systems and Transforms*, Reading, MA: Addison-Wesley, 1990.
3. A. V. Oppenhiemer and A. S. Willsky, *Signals and Systems*, 2nd ed., Upper Saddle River, NJ: Prentice Hall, 1996.
4. R. D. Strum and D. E. Kirk, *First Principles of Discrete Systems and Digital Signal Processing*, Reading, MA: Addison-Wesley, 1988.
5. C. L. Phillips, J. M. Parr, and E. A. Riskin, *Signals, Systems, and Transforms*, 4th ed., Upper Saddle River, NJ: Prentice Hall, 2008.
6. C. L. Phillips and H. T. Nagle, *Digital Control System Analysis and Design*, 3rd ed., Upper Saddle River, NJ: Prentice Hall, 1996.
7. C. L. Phillips and J. M. Parr, *Feedback Control Systems*, 5th ed., Upper Saddle River, NJ: Pearson Prentice Hall, 2011.

第4章

1. http://www.maxim-ic.com/
2. http://www.analog.com

3. H. Inose, Y. Yasuda, and J. Marakami, "A telemetering system by code modulation, delta-sigma modulation," *IRE Transactions on Space, Electronics, and Telemetry*, vol. SET-8 (September 1962): 204–209.

第5章

1. J. F. Kaiser, "Nonrecursive Digital Filter Design Using the I_0-Sinh Window Function," Proceedings of IEEE Symposium on Circuits and Systems, p. 20, 1974.
2. T. W. Parks and J. H. McClellan, "A Program for the Design of Linear Phase Finite Impulse Response Digital Filters," *IEEE Transactions on Audio and Electroacoustics*, vol. AU-20, no. 3, (August, 1072): 195–199.
3. T. W. Parks and J. H. McClellan, "Chebyshev Approximation for Nonrecursive Digital Filters with Linear Phase," *IEEE Transaction on Circuit Theory,* vol. CR-19, no. 2, (March, 1972): 189–199.
4. E. W. Cheney, *Introduction to Approximation Theory*, New York: McGraw-Hill, 1966.
5. J. H. Mathews and K. K. Fink, *Numerical Methods Using Matlab*, 4th ed., Upper Saddle River, NJ: Prentice Hall, 2004.
6. L. R. Rabiner, J. H. McClellan, and T. W. Parks, "FIR Digital Filter Design Techniques Using weighted Chebyshev Approximation," Proceedings of IEEE, no. 63, p. 595, 1975.
7. A. Antoniou, *Digital Filters, Analysis, Design, and Applications*, 2nd ed., New York: McGraw-Hill, 1993.

第6章

1. L. P. Huelsman, *Active and Passive Analog Filter Design, An Introduction*, New York: McGraw-Hill, 1993.
2. A. Antononiou, *Digital Filters Analysis, Design, and Applications*, 2nd ed., New York: McGraw-Hill, 1993.
3. M. Abramowitz and I. Stegun, *Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables*, New York: Dover, 1965.
4. M. E. Van Valkenburg, *Analog Filter Design*, New York: Holt, Rinehart, and Winston, 1980.
5. C. L. Phillips and J. M. Parr, *Signals, Systems, and Transforms*, Upper Saddle River, NJ: Prentice Hall, 1995.

第7章

1. A. Antononiou, *Digital Filters Analysis, Design, and Applications*, 2nd ed., New York: McGraw-Hill, 1993.
2. S. J. Orfanidis, *Introduction to Signal Processing*, Upper Saddle River, NJ: Prentice Hall, 1996.
3. R. J. Schilling and S. L. Harris, *Fundamentals of Digital Signal Processing*, Toronto, ON: Thompson, 2005.
4. D. Grover and J. R. Deller, *Digital Signal Processing and the Microcontroller*, Upper Saddle River, NJ: Prentice Hall, 1999.

5. J. G. Proakis and D. G. Manolakis, *Digital Signal Processing, Principles, Algorithms, and Applications*, 3rd ed., Upper Saddle River, NJ: Prentice Hall, 1996.
6. M. H. Hayes, *Statistical Digital Signal Processing and Modeling*, New York: Wiley, 1996.

第8章

1. J. G. Proakis and D. G. Manolakis, *Digital Signal Processing*, 3rd ed., Upper Saddle River, NJ: Prentice Hall, 1996.
2. E. P. Cunningham, *Digital Filtering, an Introduction*, Boston, MA: Houghton Mifflin, 1992.
3. R. A. Roberts and C. T. Mullis, *Digital Signal Processing*, Park, CA: Addison-Wesley, 1987.
4. P. M. Ebert, J. E. Mazo, and M. G. Taylor, "Overflow Oscillations in Digital Filters," *Bell System Technical Journal*, vol. 48 (November 1969): 3021–3030.
5. A. V. Oppenheim and R. W. Schafer, *Discrete-Time Signal Processing*, Upper Saddle River, NJ: Prentice Hall, 1989.

第9章

1. T. Fine, "The dawn of digital recording," *ARSC Journal*, vol 39, no. 1 (Spring 2008): 126–127.
2. M. R. Schroeder and B. F. Logan, "Colorless Artificial Reverberation," *IRE Transactions*, vol. AU-9 (1961): 209–214.
3. J. A. Moore, "About the Reverberation Business," *Computer Music Journal*, vol. 3, no. 2 (1979): 13–28.
4. J. O. Smith, *Physical Audio Signal Processing*, Palo Alto, CA: W3K Publishing, 2010.
5. U. Zolzer, *DAFX – Digital Audio Effects*, New York: Wiley, 2002.
6. S. J. Orfanidis, *Introduction to Signal Processing*, Upper Saddle River, NJ: Prentice Hall, 1996.

第10章

1. J. L. Shanks, S. Treitel, and J. H. Justice, "Stability and Synthesis of Two-Dimensional Recursive Filter," *IEEE Transactions an Audio Electroacoustics*, vol. AU-20 (June, 1972): 115–128.
2. J. S. Lim, *Two-Dimensional Signal and Image Processing*, Upper Saddle River, NJ: Prentice Hall, 1990.
3. J. S. Woods, *Multidimensional Signal, Image, and Video Processing and Coding*, Burlington, MA: Elsevier, 2006.
4. J. D. Gaskill, *Linear Systems, Fourier Transforms, and Optics*, New York: Wiley, 1978.
5. T. S. Huang, "Two-Dimensional Windows," *IEEE Transactions on Audio and Electroacoutics*, vol AU-20, (March, 1972): 88–90.
6. J. H. McClellan, "The Design of Two-Dimensional Digital Filters by Transformation," Proceedings of the 7th Annual Princeton Conference on Information Sciences and Systems, pp. 247–251, 1973.

第11章

1. A. Haar, "Zur Theorie der Orthogonalen Funktionensysteme, (Erste Mitteilung)," *Mathematische Annalen*, vol. 60 (1910): 331–371.
2. A. Grossmann and J. Morlet, "Decomposition of Hardy Functions into Square Integrable Wavelets of Constant Shape," *SIAM Journal of Mathematics*, vol. 15 (1984): 723–736.
3. I. Daubechies, "Orthonormal Bases of Wavelets with Finite Support—Connection with Discrete Filters," Proceedings of the 1987 International Workshop on Wavelets and Applications, Marseille, France, eds. J. M. Combes, A. Grossmann, and Ph. Tchamitchian, Springer, Berlin, 1989.
4. J. O. Smith, Spectral Audio Signal Processing, October 2008 Draft, http://ccrma.stanford.edu/~jos/sasp/, online book, accessed 11/20/2010.
5. I. Daubechies, *Ten Lectures on Wavelets*, Philadelphia, PA: SIAM, 1992.
6. M. J. Shensa, "The Discrete Wavelet Transform: Wedding the 'a trous' and Mallat Algorithms," *IEEE Transactions on Signal Processing*, vol. 40, no. 10 (1992): 2464–2482.
7. D. L. Fugal, *Conceptual Wavelets in Digital Signal Processing*, Spring Valley CA: Space and Signals Technical Publishing, 2009.
8. S. Mallat, *A Wavelet Tour of Signal Processing*, 2nd ed., Washington, DC: Academic Press, 1999, and D. L. Donoho, "De-noising by Soft-Thresholding," *IEEE Transactions on Information Theory*, vol. 41, no. 3 (1995): 613–627.
9. G. Strang and T. Nguyen, *Wavelets and Filter Banks*, Wellesley, MA: Wellesley-Cambridge Press, 1997.
10. S. Mallat, *A Wavelet Tour of Signal Processing*, Washington, DC: Academic Press, 1999.

附录A

1. L. P. Huelsman, *Active and Passive Analog Filter Design, An Introduction*, New York: McGraw-Hill, 1993.
2. C. L. Phillips and J. M. Parr, *Signals, Systems, and Transforms*, Upper Saddle River, NJ: Prentice Hall, 1995.
3. A. Antononiou, *Digital Filters Analysis, Design, and Applications*, 2nd ed., New York: McGraw-Hill, 1993.
4. A. Milton and S. Irene, ed., *Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables*, New York: Dover, 1965.
5. M. E. Van Valkenburg, *Analog Filter Design*, New York: Holt, Rinehart, and Winston, 1980.

附录C

1. C. L. Phillips, J. M. Parr, and E. A. Riskin, *Signals, Systems, and Transforms*, 4th ed., Upper Saddle River, NJ: Prentice Hall, 2008.

附录 C
数学背景

C.1 几何级数求和公式

1. $\sum_{m=0}^{n} a^m = \begin{cases} \dfrac{1-a^{n+1}}{1-a}, & a\neq 0,\ a\neq 1 \\ n+1, & a=1 \\ 0, & a=0 \end{cases}$

2. $\sum_{m=0}^{\infty} a^m = \dfrac{1}{1-a},\ |a|<1$

3. $\sum_{m=n}^{\infty} a^m = \dfrac{a^n}{1-a},\ |a|<1$

4. $\sum_{m=n_1}^{n_2} a^m = \dfrac{a^{n_1}-a^{n_2+1}}{1-a}$

5. $\sum_{m=0}^{\infty} ma^m = \dfrac{a}{(1-a)^2},\ a<1$

C.2 欧拉公式

$$e^{j\theta} = \cos(\theta) + j\sin(\theta)$$
$$e^{-j\theta} = \cos(\theta) - j\sin(\theta)$$
$$\cos(\theta) = \frac{e^{j\theta}+e^{-j\theta}}{2},\quad \sin(\theta) = \frac{e^{j\theta}-e^{-j\theta}}{2j}$$

C.3 双边 z 逆变换的部分分式分解

双边 z 变换收敛区域(ROC)的一般形式如图 C.1 所示。若 $F(z)$的极点在 ROC 最小圆的内部，则为右边序列。若 $F(z)$极点在 ROC 最大圆的外部，则为左边序列。

1. 利用部分分式法将 $F(z)$展开成两个函数之和

$$F(z) = F^+(z) + F^-(z) = Z\{f^+[n] + f^-[n]\}$$

式中，$F^+(z)$是右边序列的 z 变换，极点位于收敛域 ROC 最小圆的内部。而 $F^-(z)$是左边序列的变换式，它的极点位于收敛域 ROC 最大圆的外部。

2. 查询双边 z 变换表，找出 $f^+[n]$和 $f^-[n]$和其对应的 $F^+(z)$和 $F^-(z)$。

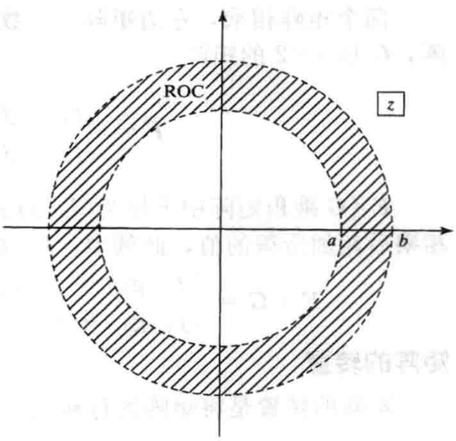


图 C.1 双边 z 变换的 ROC

例 C.3-1

求 $F(z)=\dfrac{2z(z-0.375)}{(z-0.5)(z-0.25)}$的逆变换，已知 ROC 分别为

(a) $|z|>0.5$

(b) $|z|<0.25$

(c) $0.25<|z|<0.5$

解 1. $\frac{F(z)}{z}=\frac{2(z-0.375)}{(z-0.5)(z-0.25)}=\frac{K_1}{(z-0.5)}+\frac{K_2}{(z-0.25)}$

$$K_1=\left.\frac{2(z-0.375)}{z-0.25}\right|_{z=0.5}=1.0$$

$$K_2=\left.\frac{2(z-0.375)}{z-0.5}\right|_{z=0.25}=1.0$$

$$\therefore F(z)=\frac{z}{(z-0.5)}+\frac{z}{(z-0.25)}$$

2. a. 当ROC $|z|>0.5$ 时，两个极点都在收敛域对应的圆的内部，所以 $F(z)$ 是右边序列。由 z 变换表可得，$f[n]=f^+[n]=[(0.5)^n+(0.25)^n]u[n]$。

b. 当ROC $|z|<0.25$ 时，两个极点都在收敛域对应的圆的外部，所以 $F(z)$ 是左边序列。由 z 变换表可得，$f[n]=f^-[n]=[-(0.5)^n-(0.25)^n]u[-n-1]$。

c. 当ROC $0.25<|z|<0.5$ 时，极点 $z=0.25$ 在收敛域小圆的内部，而极点 $z=0.5$ 在收敛域大圆的外部，所以 $F^+(z)=\frac{z}{(z-0.25)}$ 对应右边序列，$F^-(z)=\frac{z}{(z-0.5)}$ 对应左边序列。由 z 变换表可得，$f^+[n]=(0.25)^n u[n]$，$f^-[n]=-(0.5)^n u[-n-1]$，$f[n]=f^+[n]+f^-[n]=(0.25)^n u[n]-(0.5)^n u[-n-1]$。◀

C.4 矩阵代数运算

一般情况下，矩阵由 $M\times N$ 个元素组成，即矩阵有 M 行，N 列。

矩阵加法

两个矩阵进行加减运算时，要求两个矩阵的行数和列数相同。例如，求下列两个矩阵的和 $\boldsymbol{F}+\boldsymbol{G}$，其中

$$\boldsymbol{F}=\begin{bmatrix}f_{1,1} & f_{1,2}\\ f_{2,1} & f_{2,2}\end{bmatrix},\quad \boldsymbol{G}=\begin{bmatrix}g_{1,1} & g_{1,2}\\ g_{2,1} & g_{2,2}\end{bmatrix}$$

注意，两个矩阵中各元素的下标指明了该元素在矩阵中的行和列的位置，矩阵的和为每行、每列对应元素相加。

$$\boldsymbol{F}+\boldsymbol{G}=\begin{bmatrix}f_{1,1}+g_{1,1} & f_{1,2}+g_{1,2}\\ f_{2,1}+g_{2,1} & f_{2,2}+g_{2,2}\end{bmatrix}$$

矩阵乘法

两个矩阵相乘，左边矩阵的列数必须等于右边矩阵的行数。例如，$\boldsymbol{F}$ 是个 2×3 的矩阵，$\boldsymbol{G}$ 是 3×2 的矩阵

$$\boldsymbol{F}=\begin{bmatrix}f_{1,1} & f_{1,2} & f_{1,3}\\ f_{2,1} & f_{2,2} & f_{2,3}\end{bmatrix},\quad \boldsymbol{G}=\begin{bmatrix}g_{1,1} & g_{1,2}\\ g_{2,1} & g_{2,2}\\ g_{3,1} & g_{3,2}\end{bmatrix}$$

$\boldsymbol{F}\times\boldsymbol{G}$ 乘积矩阵中下标为(i,j)的元素，等于 $\boldsymbol{F}$ 的第 i 行元素与 $\boldsymbol{G}$ 中第 j 列元素分别相乘后相加所得的值，此例中，$\boldsymbol{F}\times\boldsymbol{G}$ 得到的是 2×2 的矩阵：

$$\boldsymbol{F}\times\boldsymbol{G}=\begin{bmatrix}f_{1,1}g_{1,1}+f_{1,2}g_{2,1}+f_{1,3}g_{3,1} & f_{1,1}g_{1,2}+f_{1,2}g_{2,2}+f_{1,3}g_{3,2}\\ f_{2,1}g_{1,1}+f_{2,2}g_{2,1}+f_{2,3}g_{3,1} & f_{2,1}g_{1,2}+f_{2,2}g_{2,2}+f_{2,3}g_{3,2}\end{bmatrix}$$

矩阵的转置

矩阵的转置是将矩阵的行和列互换，例如，求矩阵 $\boldsymbol{F}$ 的转置

$$\boldsymbol{F}=\begin{bmatrix}f_{1,1} & f_{1,2} & f_{1,3}\\ f_{2,1} & f_{2,2} & f_{2,3}\end{bmatrix}$$

得到

$$\boldsymbol{F}^{\mathrm{T}}=\begin{bmatrix} f_{1,1} & f_{2,1} \\ f_{1,2} & f_{2,2} \\ f_{1,3} & f_{3,2} \end{bmatrix}$$

这里，矩阵 $\boldsymbol{F}^{\mathrm{T}}$ 中各元素的下标仍用 $\boldsymbol{F}$ 中的下标表示，可以看出 $\boldsymbol{F}^{\mathrm{T}}$ 的第一列元素就是 $\boldsymbol{F}$ 中的第一行元素。

单位矩阵

单位矩阵是一个方阵，它的对角线上的元素均为 1，其他元素都为 0。例如，一个三阶的单位矩阵为

$$\boldsymbol{I}=\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

一个 n 阶的单位矩阵可以用 MATLAB 的命令实现

$$I=\mathrm{eye}(n)$$

矩阵与单位矩阵相乘，相当于与标量数值 1 相乘。

C.5 状态变量方程

状态变量方程的一般形式为

$$\boldsymbol{x}[k+1]=\boldsymbol{A}\boldsymbol{x}[k]+\boldsymbol{B}\boldsymbol{u}[k]$$
$$\boldsymbol{y}[k]=\boldsymbol{C}\boldsymbol{x}[k]+\boldsymbol{D}\boldsymbol{u}[k]$$

式中，$\boldsymbol{x}[k]$是表示系统内部状态的 n 个信号向量，称为系统的状态变量。该向量是一个 n 维列向量，可表示为

$$\boldsymbol{x}[k]=\begin{bmatrix} x_1[k] \\ x_2[k] \\ x_3[k] \\ \vdots \\ x_n[k] \end{bmatrix}$$

向量 u 表示 r 个输入信号，可写为由 r 个元素组成，可表示为

$$\boldsymbol{u}[k]=\begin{bmatrix} u_1[k] \\ u_2[k] \\ u_3[k] \\ \vdots \\ u_r[k] \end{bmatrix}$$

向量 $\boldsymbol{y}$ 表示 m 个输出信号，可写为由 m 个元素组成，可表示为

$$\boldsymbol{y}[k]=\begin{bmatrix} y_1[k] \\ y_2[k] \\ y_3[k] \\ \vdots \\ y_m[k] \end{bmatrix}$$

矩阵 $\boldsymbol{A}$ 称为系统矩阵，$\boldsymbol{A}$ 中各元素控制系统状态 $\boldsymbol{x}[k]$在输入信号作用下的变化。

矩阵 $\boldsymbol{B}$ 称为输入矩阵，表示系统输入向量 $\boldsymbol{u}[k]$各元素的权值。

矩阵 $\boldsymbol{C}$ 称为输出矩阵，表示系统每个状态变量对于系统输出的权值。

矩阵 $\boldsymbol{D}$ 表示输入与输出的直接耦合。

根据矩阵代数运算时的要求，如果 $\boldsymbol{x}$ 是 $n\times1$ 矩阵，$\boldsymbol{u}$ 是 $r\times1$ 的矩阵，且 $\boldsymbol{y}$ 是 $m\times1$ 的矩阵，那么，$\boldsymbol{A}$ 必须是一个 $n\times n$ 的矩阵，$\boldsymbol{B}$ 为 $n\times r$，$\boldsymbol{C}$ 为 $m\times n$，$\boldsymbol{D}$ 为 $m\times r$。

例 C.5-1

利用状态变量方程表示由下列二阶线性差分方程描述的系统

$$y[n]-0.8y[n-1]+0.4y[n-2]=2u[n-2]$$

式中，$y[n]$是输出序列，$u[n]$是输入序列。

先将方程中的 n 用 $n+2$ 代替，得到

$$y[n+2]-0.8y[n+1]+0.4y[n]=2u[n]$$

再令 $x_1[n]=y[n]$，$x_2[n]=y[n+1]=x_1[n+1]$，代入上式得

$$x_1[n+1]=x_2[n]$$

$$x_2[n+1]=-0.4x_1[n]+0.8x_2[n]+2u[n]$$

$$y[n]=x_1[n]$$

系统的数学模型用矩阵形式表示为

$$\begin{bmatrix}x_1[k+1]\\x_2[k+1]\end{bmatrix}=\begin{bmatrix}0 & 1\\-0.4 & 0.8\end{bmatrix}\begin{bmatrix}x_1[k]\\x_2[k]\end{bmatrix}+\begin{bmatrix}0\\2\end{bmatrix}u[k]$$

$$y[k]=\begin{bmatrix}1 & 0\end{bmatrix}\begin{bmatrix}x_1[k]\\x_2[k]\end{bmatrix}+[0]u[k]$$

可用 z 变换解状态方程，即将方程两边各项同时进行 z 变换。例如，一个单输入单输出的二阶系统，进行 z 变换

$$Z\begin{Bmatrix}x_1[k+1]\\x_2[k+1]\end{Bmatrix}=\boldsymbol{A}Z\begin{Bmatrix}x_1[k]\\x_2[k]\end{Bmatrix}+\boldsymbol{B}Z\{u[k]\}$$

$$Z\{y[k]\}=\boldsymbol{C}Z\begin{Bmatrix}x_1[k]\\x_2[k]\end{Bmatrix}+\boldsymbol{D}\{u[k]\}$$

得到

$$\begin{bmatrix}zX_1(z)\\zX_2(z)\end{bmatrix}=\boldsymbol{A}\begin{bmatrix}X_1(z)\\X_2(z)\end{bmatrix}+\boldsymbol{B}U(z)$$

$$\boldsymbol{Y}(z)=\boldsymbol{C}\begin{bmatrix}X_1(z)\\X_2(z)\end{bmatrix}+\boldsymbol{D}U(z)$$

利用单位矩阵重写上述方程，以将 z 从向量中分离出来，表示如下

$$[z\boldsymbol{I}-\boldsymbol{A}]\boldsymbol{X}(z)=\boldsymbol{B}U(z)$$

$$\boldsymbol{Y}(z)=\boldsymbol{C}\boldsymbol{X}(z)+\boldsymbol{D}U(z)$$

从而求得 $\boldsymbol{X}(z)$和 $\boldsymbol{Y}(z)$为

$$\boldsymbol{X}(z)=[z\boldsymbol{I}-\boldsymbol{A}]^{-1}\boldsymbol{B}U(z)$$

$$\boldsymbol{Y}(z)=[C[z\boldsymbol{I}-\boldsymbol{A}]^{-1}\boldsymbol{B}+\boldsymbol{D}]U(z)$$

所得最终结果具有一般性，适用于任意阶系统，任意数量的输入与输出。对于单输入、单输出的情况，系统传输函数为

$$\frac{Y(z)}{U(z)}=\boldsymbol{C}[z\boldsymbol{I}-\boldsymbol{A}]^{-1}\boldsymbol{B}+\boldsymbol{D}$$

◀

参考文献[1]给出了关于状态变量方程更深入的介绍。

附录 D

MATLAB 用户自定义函数和命令

D.1 MATLAB 用户自定义函数

下面介绍的 MATLAB 自定义函数主要是本书中使用到的函数，其并不是 Mathworks 发布的函数。

Waveform Generating Functions(波形发生函数)

Band-Limited Square Wave(带限方波)

```
%fouriersq.m - Generates a square wave from the Fourier series
% using a fixed number of points
%
% sqwave = fouriersq(11025, 1024, 100, 15);
%
% Evaluates the Fourier expansion for a square wave
% fs - sample frequency
% pts - number of points in the wave
% f0 - base frequency of the square wave
% terms - number of terms in the Fourier sum.
% Maximum frequency is (2*terms - 1)*f0.
   function sq = fouriersq(fs, pts, f0, terms)
   T = 1/fs;
   n = 1:pts;
   nT = n*T;
   w = 2*f0*pi;
   sq = zeros(1, length(nT));
   for k = 1:2:2*terms
       sq = sq + (4/(k*pi))*sin(k*w*nT);
   end
```

Band-Limited Triangular Wave(带限三角波)

```
% fouriertri.m - Generates a triangular wave from the Fourier
  series using
% a fixed number of points
%
% triwave = fouriertri(11025, 1024, 100, 15);
%
% Evaluates the Fourier expansion for a triangular wave
% fs - sample frequency
% pts - number of points in the wave
% f0 - base frequency of the square wave
% terms - number of terms in the Fourier sum.
% Maximum frequency is (2*terms -1)*f0.

   function tr = fouriertri(fs, pts, f0, terms)
   T = 1/fs;
```

```
n = 1:pts;
nT = n*T;
w = 2*f0*pi;
tr = zeros(1, length(nT));
for k = 1:2:2*terms
    tr = tr + (8/(k*pi)^2)*cos(k*w*nT);
end
```

Band-Limited Sawtooth Wave(带限锯齿波)

```
% fouriersaw.m - Generates a sawtooth wave from the Fourier series
% using a fixed number of points
%
% sawwave = fouriersaw(11025, 1024, 100, 15);
%
% Evaluates the Fourier expansion for a sawtooth wave
% fs - sample frequency
% pts - number of points in the wave
% f0 - base frequency of the square wave
% terms - number of terms in the Fourier sum.
% Maximum frequency of the sawtooth is terms*f0.

  function saw = fouriersaw(fs, pts, f0, terms)
  T = 1/fs;
  n = 1:pts;
  nT = n*T;
  w = 2*f0*pi;
  saw = zeros(1, length(nT));
  for k = 1:terms
      saw = saw + 1/(2*k*pi)*sin(k*w*nT);
  end
```

Quantization Functions(量化函数)

Quantize to B-bits(量化为 B 位)

```
%quant.m - Quantizes a signal x to B-bits.
% x = .97634;
% xq = quant(x, 4, -1, 1);
%
% Changes the representation of a signal x to a signal xq which is a
% quantized version of x.
% xq is limited to B-bits and must fall between min <= x < max.
% If min and max are not specified they are taken to be the range
% -1 <= x < 1
  function s = quant(xd, B, min, max)
%Converts xd to a quantized number using B bits.
% min <= xd < max
  if ~(nargin == 2 || nargin == 4)
    error('usage: quant(x, BIT, MIN, MAX).');
  end;
  if B <= 0
    error('Bits must be an integer > 0');
  end;
  if nargin == 2
    min = -1;max = 1;
  end
```

```
x = xd - min;
x = x/(max - min);
x = fix(x*(2^B));                   % Make x a whole number
x = x/(2^B);
s = x*(max - min) + min;
```

Two's Complement(二进制补码)

```
%twosover.m - Applies the rules of two's complement
% overflow to a signal x.
% y = 1.1;
% x = twosover(y);
%
% if -1 <= x < 1 this function returns x.
% if x < -1 it returns x + 2
% if x >= 1 it returns x - 2

function s = twosover(x)
 x(x >= 1) = x(x>= 1) - 2;
 x(x < -1) = x(x < -1) + 2;
 s = x;
 return;
```

Plotting Functions(图形函数)

3D Plot(3维图形)

```
%plot3d.m - plots an image in 3d using surf
%
% plot3d(im, N, fignum, imap, viewpt, axis3d);
%
% im is the image
% N is the size - assumes N x N
% fignum is the figure number
% imap is the shading map
% viewpt is 1 x 2 and gives [angle elevation]
% axis3d is 1 x 4 and gives the axis dimensions.
%
% viewpt and axis3d are optional parameters.
% viewpt defaults to [0 90]
% axis3d defaults to [0 N-1 0 N-1 -2 2]
   function plot3d(im, N, fignum, imap, viewpt, axis3d)
     if nargin < 6, axis3d = [0 N-1 0 N-1 -2 2]; end
     if nargin < 5, viewpt = [0 90]; end
     az = viewpt(1, 1);el = viewpt(1, 2);
     figure(fignum);clf;
     surf(im);
     shading flat;
     colormap(imap);
     axis(axis3d);
     view(az, el);
     xlabel('n1');
     ylabel('n2');
     grid off;
```

Plot FFT for an Image(图像的FFT变换结果图)

```
%plotfft2.m - Plots the image in 3D, a top view of the image, and
%               the fft of the image on a single figure with
%               three subplots
%
% Usage: plotfft2(im, N, fignum, imap)
%
% im - the image
% N - assumes N x N
% fignum - figure number to be assigned
% imap - the color map
function plotfft2(im, N, fignum, imap)
    if(nargin == 3)
      imap = colormap(jet(128));
    end
    if(nargin == 2)
      figure;clf;
    else
      figure(fignum);clf;
    end
    az = 0;
    el = 90;
    y = fft2(im);
    ymag = abs(y);
    ymag = ymag/abs(max(max(ymag)));
    [x y] = meshgrid([0:N-1]);
    subplot(1, 3, 1);
    surf(im);
    colormap(imap);
    shading interp;
    axis([0 N-1 0 N-1 -.2 1.2]);
    xlabel('n1');
    ylabel('n2');
    subplot(1, 3, 2);
    surf(im);
    colormap(imap);
    shading flat;
    axis([0 N-1 0 N-1 -2 2]);
    view(az, el);
    xlabel('n1');
    ylabel('n2');
    subplot(1, 3, 3);
    surf(fftshift(ymag));
    colormap(imap);
    axis([0 N-1 0 N-1 0 2]);
    view(az, el);
    shading flat;
    xlabel('n1');
    ylabel('n2');
```

Invert the Color Map(彩色图反色)

```
%invertcolormap - Inverts the color scale. Use to change a figure
% that is mostly black to one that is mostly white
%
%imap = invertcolormap(colormap(gray(128));
%
    function imap = invertcolormap(map)
    imap = zeros(size(map));
    for i = 1:size(map, 1);
        imap(i, 1:3) = map(size(map,1)-i+1, 1:3);
    end
```

Reverberation Functions(混响函数)

Simple Reverberator(简单混响器)

```
%reverbsimple.m
%
% [y num den] = reverbsimple(xin, delay, gain);
% or y = reverbsimple(xin, delay, gain);
%
% Creates a simple reverberator with gain A and delay d with
% transfer function 1/(z^d-A) and uses MATLAB's filter function
% to filter the input vector.
%
    function [yout b a] = reverbsimple(u, d, A)
    num = 1;
    den = [1 zeros(1, d-1) -A];
    y = filter(num, den, u);
    if(nargout == 1)
      yout = y;
    else
      yout = y;
      b = num;
      a = den;
    end
```

All-Pass Reverberation Filter(全通混响滤波器)

```
%reverballpass.m
%
% [y num den] = reverballpass(xin, delay, gain);
% or y = reverallpass(xin, delay, gain);
%
% Creates an all pass reverberator of gain A and delay d with
% transfer function -A(z^d -1/A)/(z^d-A) and uses
% MATLAB's filter function to filter the input vector
%.
function [yout b a] = reverballpass(u, d, A)
num = -A*[1 zeros(1, d-1) -1/A];
den = [1 zeros(1, d-1) -A];
y = filter(num, den, u);
if(nargout == 1)
```

```
      yout = y;
   else
      yout = y;
      b = num;
      a = den;
   end
```

Low-Pass Reverberation Filtering(低通混响滤波)

```
%reverblowpass.m
%
% [y num den] = reverblowpass(xin, delay, gain, realpole, realzero);
% y = reverblowpass(xin, delay, gain, realpole, realzero);
%
% Creates a low pass filter reverberator of gain A and delay d
% with a real pole at realpole and a real zero at realzero.
% It uses MATLAB's filter function to filter the input vector.
    function [yout b a] = reverblowpass(u, d, A, p1, z1)
    K = (1-p1)/(1-z1);
    num = A*[1 -p1 zeros(1, d)];
    den = [1 -p1 zeros(1, d-2) -A*K*p1 A*K*z1];
    y = filter(num, den, u);
    if(nargout == 1)
       yout = y;
    else
      yout = y;
      b = num;
      a = den;
    end
```

Equalizer Functions(均衡函数)

First-Order Equalizer(一阶均衡器)

```
function [num, den] = equalizer1(low, A, fs, fc, Ac)
%equalizer1.m - Creates a first order parametric equalizer
%
% [num den] = equalizer(low, A, fs, fc, Ac);
%
% low - Determines low or high filter
%       low = 0 for a high pass shelving filter
%       low = 1 for a low pass shelving fitler
% A - the gain for the boost or the cut
%       A > 1 for a boost filter
%       0 < A < 1 for a cut filter
% fs - sample frequency in Hz
% fc - the cutoff frequency
% Ac - the gain at fc. If Ac is omitted the default value is
%       Ac = (sqrt(2)/2)*(A - 1) + 1 for a boost
%       Ac = (sqrt(2)/2)*(1 - !) + A for a cut
%
if (nargin < 4 || nargin > 5)
   disp('error');
```

```
    disp('Usage: [num den] = equalizer1(low, A, fs, fc, Ac);');
    disp('Usage: [num den] = equalizer1(low, A, fs, fc);');
    return;
end
if(nargin == 4 && A > 1)
    Ac = (sqrt(2)/2)*(A - 1) + 1;
end
if(nargin == 4 && A < 1)
    Ac = 1 - (sqrt(2)/2)*(1 - A);
end
alpha = pi*fc/fs;
if (low == 1)
    d = sqrt((Ac^2 - 1)/(A^2 - Ac^2));
else
    d = sqrt((Ac^2 - A^2)/(1 - Ac^2));
end
p1 = (1 - d*tan(alpha))/(1 + d*tan(alpha));
beta = (1 - p1)/(1 + p1);
if(low == 1)
    z1 = (1 - A*beta)/(1 + A*beta);
    K = (1 + p1)/(1 + z1);
else
    z1 = (A - beta)/(A + beta);
    K = (1 - p1)/(1 - z1);
end
num = K*[1 -z1];
den = [1 -p1];
return;
```

Second-Order Equalizer(二阶均衡器)

```
function [num, den] = equalizer2(f0, A, bw, fs, Abw)

%equalizer2.m - Creates a second order parametric equalizer
%
% [num den] = equalizer(f0, A, bw, fs, Abw);
%
% f0 - center frequency
%      f0 = 0 for a low pass shelving filter
%      f0 = fs/2 for a high pass shelving fitler
%      0 < f0 < fs/2 creates a peak or notch filter
% A - the peak or notch gain
%     A > 1 for a peak or boost filter
%     0 < A < 1 for a notch or cut filter
% bw - bandwidth in Hz
% fs - sample frequency in Hz
% Abw - the gain at bandwidth frequency
%       Abw defaults to A - (A-1)/sqrt(2)
if(nargin < 4 || nargin > 5)
    disp('error');
    disp('Usage: [num den] = equalizer(f0, A, bw, fs, Abw);');
    disp('       [num den] = equalizer(f0, A, bw, fs);')
```

```
        return;
    end
    if(nargin == 4)
        Abw = A - (A-1)/sqrt(2);
    end
    f0 = f0/fs;w0 = 2*pi*f0;
    bw = 2*pi*(bw/fs);
    B = sqrt((Abw^2 - 1)/(A^2 - Abw^2))*tan(bw/2);
    a0 = (1 + A*B);
    a1 = -2*cos(w0);
    a2 = 1 - A*B;
    b0 = 1 + B;
    b1 = -2*cos(w0);
    b2 = 1 - B;
    num = [a0 a1 a2]/b0;
    den = [b0 b1 b2]/b0;
    return;
```

WAV File Plotting Functions(WAV 格式数据绘图函数)

Plot a WAV File in Time and Frequency(画 WAV 格式数据的时间和频率图)

```
%plotwav.m
%
% plotwav('wavfile');
%
% Reads a wave file and displays two subplots, one in the time
% domain and the other in the frequency domain, for each channel.
%
    function plotwav(WavName)
    if(nargin ~= 1)
        disp('Error');
        disp('Usage: PlotWav(wavefilename)');
        return;
    end
    [u fs b] = wavread(WavName);
    T = 1/fs;
    k = 1:length(u);
    s = size(u, 2);
    for i = 1:s
        figure;clf;
        subplot(2,1,1)
        x = u(:, i);
        plot(k*T,x)            %Plot x in time
        axis([0 T*length(u) -1.5 1.5])
        xlabel('time in seconds');
        ylabel('voltage');
        title([WavName 'channel' num2str(i)]);
        %
        U = fft(x);
        U = U/max(abs(U));
        subplot(2,1,2)
```

```
    plot(k*fs/length(u), abs(U)) %Plot U in frequency
    axis([0 fs/2 0 1]);
    xlabel('frequency in Hz');
    ylabel('gain');
end
return;
```

Plot Two WAV Files in Frequency for Comparison(绘制2个WAV格式数据波形的频域图，并进行比较)

```
%plot2wav.m
%
% plot2wav('wavfile1', 'wavfile2');
%
% Reads two wave file and displays the frequency plot of each.
% Multiple channels are put on multiple figures.
%
function plot2wav(WavName1, WavName2)
if(nargin ~= 2)
   disp('Error');
   disp('Usage: plot2wav(wavefile1, wavefile2)');
   return;
end
[u1 fs b] = wavread(WavName1);
[u2 fs2 b2] = wavread(WavName2);
s1 = size(u1, 2);s2 = size(u2, 2);
if(s1 ~= s2)
   disp('Error');
   disp('Both wavfiles must have the same number of channels');
   return;
end
for i = 1:s1
    figure;clf;
    k = 1:length(u1);
    subplot(2,1,1)
    U1 = fft(u1(:, i));
    U1 = U1/max(abs(U1));
    plot(k*fs/length(u1), abs(U1)) %Plot U1 in frequency
    axis([0 fs/2 0 1]);
    xlabel('frequency in Hz');
    ylabel('gain');
    title([WavName1 'channel' num2str(i)]);
    subplot(2,1,2)
    k2 = 1:length(u2);
    U2 = fft(u2(:,i));
    U2 = U2/max(abs(U2));
    plot(k2*fs2/length(u2), abs(U2)); %Plot u2
    axis([0 fs2/2 0 1])
    xlabel('frequency in Hz');
    ylabel('gain');
    title([WavName2 'channel ' num2str(i)]);
```

```
end
    %
return;
```

D.2 MATLAB命令

下面介绍的是本书例题中使用的MATLAB命令。每个命令的描述都是相应的MATLAB帮助文件中的删减版。完整的描述和举例可以通过在MATLAB的命令行中输入help<*command name*>得到。

blackman：布莱克曼窗函数。

- blackman(N)以列向量的形式返回 N 点对称布莱克曼窗。

butter：巴特沃斯数字和模拟滤波器设计。

- [B，A]=butter(N，Wn)用于设计 N 阶的低通数字巴特沃斯滤波器。该函数返回滤波器的系数，即长度为 $N+1$ 的 A(分母)和 B(分子)两个向量，系数按照 z 的降幂顺序排列。截止频率Wn的范围是0.0<Wn<1.0，1.0对应抽样频率的一半。
- [B，A]=butter(N，Wn，'high')用于设计 N 阶的高通数字巴特沃斯滤波器。其中，'high'可改为'stop'或'pass'，分别表示带阻滤波器和带通滤波器，对于带通和带阻滤波器，Wn是二维向量，对应上限截止频率和下限截止频率。
- [B，A]=butter(N，Wn，'s')用于设计 N 阶低通模拟滤波器。Wn是截止频率，单位为rad/s。

buttord：巴特沃斯滤波器阶数选择函数。

- [N，Wn]=buttord(Wp，Ws，Rp，Rs)返回数字巴特沃斯滤波器的最小阶数，该滤波器满足通带内的衰减不高于Rp dB，阻带内的衰减不低于Rs dB。Wp和Ws分别是通带、阻带的截止频率，归一化范围为[0，1](1对应于pi弧度/抽样点)。
- [N，Wn]=buttord(Wp，Ws，Rp，Rs，'s')返回模拟滤波器的阶数 N。

ceil：向正无穷大方向取整函数。

- ceil(X)返回 X 中每个元素在正无穷方向上的最靠近该元素的整数。

cell：构建单位数组。

- cell(N)构建一个 $N\times N$ 的单元列矩阵，每个矩阵为空矩阵。

cheb1ord：切比雪夫Ⅰ型滤波器阶数选择函数。

- [N，Wp]=cheb1ord(Wp，Ws，Rp，Rs)返回数字切比雪夫Ⅰ型滤波器的最小阶数 N，满足在通带内的衰减不高于Rp dB，而在阻带内的衰减不低于Rs dB。Wp和Ws分别是通带、阻带的截止频率，归一化范围为[0，1](1对应于pi弧度/抽样点)。
- [N，Wp]=cheb1ord(Wp，Ws，Rp，Rs，'s')用于设计模拟滤波器。

cheb2ord：切比雪夫Ⅱ型滤波器阶数选择函数。

- [N，Ws]=cheb2ord(Wp，Ws，Rp，Rs)返回数字切比雪夫Ⅱ型滤波器的最小阶数 N，满足通带内的衰减不高于Rp dB，而在阻带内的衰减不低于Rs dB。Wp和Ws分别是通带、阻带的截止频率，归一化范围为[0，1](1对应于pi弧度/抽样点)。
- [N，Ws]=cheb1ord(Wp，Ws，Rp，Rs，'s')用于设计模拟滤波器。

cheby1：切比雪夫Ⅰ型数字和模拟滤波器设计。

- [B，A]=cheby1(N，R，Wp)用于设计 N 阶低通数字切比雪夫滤波器，R 是指通带内峰峰值间波纹的dB值。该函数返回滤波器的系数，即长度为 $N+1$ 的 A(分母)和 B(分子)两个向量。Wp为通带截止频率，且0.0<Wp<1.0，1.0表示抽样频率的一半。

- [B，A]＝cheby1(N，Wn，'high')用于设计 N 阶的高通数字切比雪夫滤波器。表达式中'high'可改为'stop'或'pass'，分别表示带阻滤波器和带通滤波器。当为带阻或带通滤波器时，Wn是二维向量，对应上限截止频率和下限截止频率。
- [B，A]＝cheby1(N，Wn，'s')用于设计 N 阶低通模拟滤波器。Wn是截止频率，单位为rad/s。

cheby2：切比雪夫Ⅱ型数字和模拟滤波器设计。

- [B，A]＝cheby2(N，R，Wst)用于设计 N 阶低通数字切比雪夫滤波器，R 是指阻带波纹系数的dB表示。该函数返回滤波器的系数，即长度为 $N+1$ 的 A(分母)和 B(分子)两个向量。Wst为阻带截止频率，且0.0<Wst<1.0，1.0表示抽样频率的一半。
- [B，A]＝cheby2(N，Wn，'high')用于设计 N 阶的高通数字切比雪夫滤波器。表达式中'high'可改为'stop'或'pass'，分别表示带阻滤波器和带通滤波器。当为带阻或带通滤波器时，Wn是二维向量，对应上限截止频率和下限截止频率。
- [B，A]＝cheby2(N，Wn，'s')用于设计 N 阶低通模拟滤波器。Wn是截止频率，单位为rad/s。

chirp：产生余弦扫频信号。

- Y＝chrip(T，F0，T1，F1)产生线性扫频信号，时刻值由数组 T 定义。时刻0的瞬时频率为 $F0$Hz，时刻 $T1$ 的瞬时频率为 $F1$Hz。默认情况下，$F0=0$，$T1=1$，$F1=100$。

conv：卷积和多项式相乘运算函数。

- C＝conv(A，B)表示 A 和 B 的卷积。卷积结果的向量长度为MAX([LENGTH(A)＋LENGTH(B)－1，LENGTH(A)，LENGTH(B)])。如果 A 和 B 是两个多项式的系数，卷积的结果等效于两个多项式的乘积。

dct：离散余弦变换函数。

- Y＝dct(X)返回序列 X 的离散余弦变换结果 Y，Y 的长度与 X 的长度相同，其元素表示离散余弦变换系数。
- Y＝DCT(X，N)将序列 X 补零或截断到长度 N 后再进行离散余弦变换。

dct2：二维离散余弦变换函数。

- B＝dct2(A)返回矩阵 A 的离散余弦变换结果。矩阵 B 与矩阵 A 大小相同，B 中的元素表示离散余弦变换系数。

disp：显示数组。

- disp(X)显示一个数组的内容，但不会显示数组的名称。除了空数组不能显示外，在其他方面与表达式后面去掉分号的功能一样。

downsample：对输入信号降采样。

- downsample(X，N)从输入信号 X 的第一项开始，等间隔 N 对其采样。如果 X 是矩阵，则对 X 的每一列进行采样。

ellip：椭圆数字和模拟滤波器设计。

- [B，A]＝ellip(N，Rp，Rs，Wp)用于设计 N 阶低通数字椭圆滤波器，Rp为通带峰峰值间的波纹系数的dB值，Rs为阻带衰减的dB值。
- [B，A]＝ellip(N，Wn，Rp，Rs，'high')用于设计 N 阶的高通数字椭圆滤波器。表达式中'high'可改为'stop'或'pass'，分别表示带阻滤波器和带通滤波器。当为带阻或带通滤波器时，Wn是二维向量，对应上限截止频率和下限截止频率。
- [B，A]＝ellip(N，Wn，Rp，Rs，'s')用于设计 N 阶低通模拟滤波器。Wn是截止频率，单位为rad/s。

elliport：椭圆滤波器阶数选择函数。

- [N，Wp]=ellipord(Wp，Ws，Rp，Rs)返回数字椭圆滤波器的最小阶数 N，满足在通带内的衰减不高于 Rp dB，而在阻带内的衰减不低于 Rs dB。Wp 和 Ws 分别是通带和阻带的截止频率，范围为[0，1](1 对应于 pi 弧度/抽样点)。
- [N，Wp]=ellipord(Wp，Ws，Rp，Rs，'s')返回模拟椭圆滤波器的最小阶数。

fft：离散傅里叶变换。

- fft(X)用于计算序列向量 X 的离散傅里叶变换(DFT)，若 X 为矩阵，则是对 X 的每一列进行 DFT。
- fft(X，N)计算 X 的 N 点 DFT，如果 X 的长度小于 N，则先对其补零。

fft2：二维离散傅里叶变换。

- fft2(X)返回矩阵 X 的二维 DFT，若 X 为一个向量，则与 fft 的计算结果相同。
- fft2(X，MROWS，NCOLS)是先将矩阵 X 补零成一个 MROWS×NCOLS 的矩阵，再进行变换。

fftshift：将零频分量移至频谱中心的函数。

- 当 X 为向量时，fftshift(X)把 X 的左右两半进行交换。如果 X 为矩阵，则同时把 X 的第一象限和第三象限，第二象限和第四象限交换。

filter：一维数字滤波函数。

- Y=filter(B，A，X)利用 A，B 向量描述的滤波器对输入向量 X 进行滤波处理，产生输出序列向量 Y。A 和 B 分别为滤波器系统函数的分母和分子多项式系数向量。滤波器是标准差分方程的直接 2 型转置(direct form 2 transposed)。

filter2：二维数字滤波器。

- Y=filter2(B，X)采用矩阵 B 描述的二维 FIR 滤波器对输入序列 X 滤波，使用二维相关计算输出结果 Y，Y 的大小与 X 一致。

fir1：用窗函数法设计 FIR 滤波器。

- B=fir1(N，Wn)用于设计 N 阶的低通 FIR 数字滤波器，返回的滤波器系数为长度为 $N+1$ 的向量 B。Wn 为截止频率，且 0.0<Wn<1.0，1.0 表示抽样频率的一半。函数返回的 B 是一个实数，且有线性相位。滤波器在截止频率 Wn 上的归一化增益为−6dB。
- B=fir1(N，Wn，WIN)使用 WIN 指定的窗函数设计滤波器，如果 WIN 没有给出特定的窗函数，则自动取汉明窗。此函数还可以设计带通、带阻、高通和多通道滤波器，详见 MATLAB 的 help 文件。

fir2：基于频率抽样的任意形状FIR滤波器设计。

- B=fir2(N，F，A)用于设计 N 阶线性相位 FIR 数字滤波器，滤波器的频率响应由向量 F 和 A 指定。返回的滤波器系数是长度为 $N+1$ 的向量 B。向量 F 和 A 分别指定所需滤波器抽样点的频率和幅值。F 中各频率按递增顺序排列，且 $0.0<F<1.0$，1.0 表示抽样频率的一半。

firls：基于最小均方误差的线性 FIR 滤波器设计。

- B=firls(N，F，A)返回长度为 $N+1$ 的线性相位(实对称系数)滤波器，该滤波器是 F 和 A 描述的频率响应在最小均方意义下的最佳逼近。F 的元素是成对的频带边缘，按升序排列，取 0~1.0之间的值，1 是奈奎斯特频率，或者为采样频率的一半。A 跟 F 的大小相同，是一个实向量，指定滤波器 B 中频率点的频率响应幅值。

firpm：帕克斯-麦克莱伦优化等波纹 FIR 滤波器设计。

- B=firpm(N，F，A)返回长度为 $N+1$ 的线性相位(实对称系数)滤波器，该滤波器是 F 和 A 描述的频率响应在最小最大化意义下的最佳逼近。F 的元素是成对的

频带边缘，按升序列取0～1.0之间的值，1是奈奎斯特频率，或为采样频率的一半。至少有一个频带宽度不为零。*A*跟*F*的大小相同，是一个实向量，指定滤波器*B*中频率点的频率响应幅值。

firpmord：帕克斯-麦克莱伦优化等波纹FIR滤波器阶数估计。

- [N，Fo，Ao，W]=firpmord(F，A，DEV，Fs)得到近似的阶数*N*，归一化频带边缘Fo、频带幅值Ao和权值*W*，这些参数可供firpm函数使用，firpm函数调用firpmord返回的参数的格式为：B=firpm(N，Fo，Ao，W)，返回的滤波器满足*F*、*A*、DEV给定的参数指标。*F*是截止频率(Hz)，按升序列排列，在0～Fs/2之间取值，Fs是采样频率。

fix：向0方向取最近整数函数。

- fix(X)返回*X*中每个元素沿0方向的最近整数。

floor：向负无穷方向取最近整数函数。

- floor(X)返回*X*中每个元素沿负无穷方向最近的整数。

freqs：拉普拉斯变换(s域)频率响应函数。

- H=freqs(B，A，W)返回滤波器的复频率响应*H*，该滤波器系统函数由分母*B*和分子*A*决定，*H*是向量*W*(rad/s)指定频率点上的频率响应。当没有输出向量*H*时，freqs(B，A，W)函数将绘出频率响应的幅频和相频曲线。

freqz：数字滤波器频率响应函数。

- [H，W]=freqz(B，A，N)返回滤波器的*N*点复频率响应向量*H*和*N*点频率向量*W*，*W*的单位为弧度/抽样点。滤波器系统函数由分母*B*和分子*A*决定。频率响应由均匀分布在单位圆上半平面的*N*个频点求解，若没有给出*N*的特定值，则默认$N=512$。
- [H，W]=freqz(B，A，N，fs)允许指定抽样频率。

freqz2：二维频率响应函数。

- [H，Fx，Fy]=freqz2(h，Nx，Ny)返回h对应的Nx×Ny频率响应*H*，频率向量Fx(长度为*Nx*)和频率向量Fy(长度为*Ny*)。*h*是二维FIR滤波器，以微分形式表示。返回的Fx和Fy是−1.0～1.0之间的归一化频率，1.0对应抽样频率的一半。

fsamp2：基于抽样频率的二维FIR滤波器设计。

- fsamp2基于笛卡儿域上抽样点描述的所需二维频率响应设计FIR滤波器。
- H=fsamp2(HD)使用频率响应HD设计二维FIR滤波器，调用函数后返回滤波器的系数矩阵*H*。HD是一个矩阵，包含*x*和*y*频率轴上−1.0～1.0之间等间隔抽样点的频率响应，1.0对应抽样频率的一半。

ftrans2：基于频率变换法的二维FIR滤波器设计。

- H=ftrans2(B，T)产生二维FIR滤波器*H*，该二维滤波器是将一维FIR滤波器*B*进行*T*变换后得到。变换矩阵*T*内的元素决定频率的转换系数。

fwind2：基于二维窗函数的二维FIR滤波器设计。

- fwind2使用窗函数法设计二维FIR滤波器。fwind2(HD)根据给定的滤波器频率响应HD，使用二维窗函数来设计FIR滤波器，fwind2中窗函数是二维的，而fwind1中窗函数则是一维的。

hamming：汉明窗。

- hamming(N)以列向量形式返回*N*点对称的汉明窗。

hann：汉宁窗。

- hann(N)以列向量形式返回*N*点对称的汉宁窗。

ifft2：二维离散傅里叶反变换函数。

- ifft2(F)返回矩阵 F 的二维傅里叶反变换。如果 F 是一个向量，结果与ifft结果相同。
- ifft2(F，MROWS，NCOLS)先对矩阵 F 补零，使其成为MROWSxNCOLS的矩阵，再进行逆变换。

ifftshift：傅里叶反变换移频函数。

- 当 X 是向量时，ifftshift(X)把 X 的左右两半进行交换。当 X 是矩阵时，ifftshift(X)把 X 的第一象限和第三象限，第二象限和第四象限进行交换。对于一个 N 维的数组，ifftshift(X)将 X 每一维对应的空间进行对半交换。ifftshift是fftshift的逆过程。

imagesc：将数据尺度化并显示为图像。

- imagesc(...)与函数image(...)的唯一区别在于，imagesc函数将数据尺度化到整个亮度范围。

imread：图像数据读取函数。

- A=imread(FILENAME，FMT)从字符串FILENAME指定的文件中读取图像，图像可以为灰度图像或彩色图像。如果文件不在当前目录或当前MATLAB路径中，则要指明完整的路径名称。字符串FMT指定图像文件格式，即标准文件扩展名。

imshow：图像显示函数。

- imshow(I，[LOW HIGH])显示灰度图像I，指定的灰度范围为[LOW HIGH]。
- imshow(RGB)显示真彩色RGB图像。
- imshow(BW)显示二值图像，像素值为0的点显示为黑色，像素值为1的点显示为白色。

imwrite：把图像数据写入文件函数。

- imwrite(A，FILENAME，FMT)将图像A写入指定的文件中，文件名由FILENAME指定，文件格式由FMT指定。

Kaiser：凯瑟窗。

- W=Kaiser(N)以列向量形式返回 N 点的凯瑟窗 W。

mesh：三维曲面图。

- mesh(X，Y，Z，C)绘制由四个矩阵所指定的彩色网格曲面图。

mod：求余函数。

- 当y不为0时，mod(x，y)的结果为 $x-n.*y$，其中 $n=\text{floor}(x./y)$。如果 y 不是整数且比值 $x./y$ 在整数的舍入误差之内，则 n 仍为 $\text{floor}(x./y)$。输入的 x 和 y 必须是长度相同的实序列或实数。

polyval：多项式求解函数。

- Y=polyval(P，X)返回多项式 P 在 X 处的值。P 是长度为 $N+1$ 的向量，P 中的元素是按降幂顺序排列的多项式的系数。

rand：产生均匀分布的伪随机数函数。

rectwin：矩形窗。

- W=rectwin(N)返回 N 点矩形窗。

residue：部分分式展开函数。

- [R，P，K]=residue(B，A)返回用 $B(s)/A(s)$ 表示的部分分式的余数、极点和直接项。如果没有多重根，向量 B 和 A 的元素分别是分子和分母按 s 的降幂排列的系数。函数返回的 R 和 P 是列向量，分别表示余数和极点位置。直接项 K 以行向量表示。

residuez：Z 变换的部分分式展开函数。

- [R，P，K]=residue(B，A)返回用 $B(z)/A(z)$ 表示的部分分式的余数，极点和直接项。其中，向量 B 和 A 的元素分别是分子和分母按 z^(−1)的升幂排列的系数。函数返回的 R 和 P 是列向量，分别表示余数和极点位置。直接项 K 以行向量表示。

rgb2gray：通过消除色调和饱和度分量，保持亮度不变的方法将 RGB 图像转换成灰度图像。

roud(X)：返回 X 中每个元素最靠近的整数。

spectrogram：计算短时傅里叶变换(STFT)。

- S=spectrogram(X)返回信号 X 的频谱图 S。默认情况下，X 将分成 8 段，相邻段之间有 50%的重叠，每段使用汉明窗函数加窗。用于计算离散傅里叶变换的频率点数为最大值 256，或者大于每段长度的下一个 2 的幂次。如果 X 不能被平分成 8 段，则会做截断处理。

sum：求和函数。

- S=sum(X)是对向量 X 中的元素进行求和。如果 X 是一个矩阵，则对 X 每一列分别求和，S 为行向量。对于 N 维数组，sum(X)在第一个 size 大于 1 的维度上运算。如果 X 的元素是单精度或双精度浮点数，则 S 按照 X 的数据类型累加，且求和结果 S 与 X 类型相同。如果 X 不是浮点数，则 S 的累加和最终结果均为双精度数据类型。
- S=sum(X，DIM)按 DIM 指定维数求和。

surf：三维曲面图函数。

- surf(X，Y，Z，C)绘制由四个矩阵所指定的彩色曲面图，可使用 VIEW 指令指定视角。轴的刻度由 X、Y 和 Z 的范围决定，也可使用 AXIS 指令定义轴的刻度。颜色尺度化由 C 指定，或者使用 CAXIS 指令设定。尺度化的色彩使用 COLORMAP 指令设定。阴影的模式由 SHADING 指令设定。

tf2sos：传输函数到二次分式转换函数。

- [SOS，G]=tf2sos(B，A)返回 $H(z)$表示的系统的二次分式形式的矩阵 SOS 以及增益系数 G，$H(z)$的分子为 B，分母为 A。$H(z)$的零极点一定是共轭复数对。

unwrap：相位角展开函数。

- unwrap(P)函数对弧度相位 P 进行展开，其将跳变绝对值大于或等于 pi 的相位改变到其 2＊pi 的补集。其沿着 P 的第一个维度展开，且不改变该维度的第一个相位值。P 可以是标量、向量、矩阵或 N 维数组。
- unwrap(P，TOL)在展开时，用设定的跳变容值 TOL 代替默认的 pi。

vander：范德蒙矩阵。

- A=vander(V)返回范德蒙矩阵，矩阵 A 的列是向量 V 的幂，算法为 $A(i,j)=v(i)\hat{}(n-j)$。

wavplay：使用 Windows 音频播放器播放声音文件。

- wavplay(Y，FS)将信号向量 Y 以抽样频率 FS Hz 传送给 Windows 的音频播放设备。标准的音频抽样频率是 8000、11025、22050、44100Hz。矩阵 Y 中所包含的音频数据必须存储为以下数据类型：双精度型、16 位整数型或者 8 位整数型。其中，双精度数据的范围需要满足 $-1.0\leqslant y\leqslant 1.0$，超过这一范围的数据被截断。

wavread：读出 .wav 声音文件。

- Y=wavread(FILE)读取由字符串 FILE 指定的声音文件，将采样数据返回到 Y。如果没有指定扩展名，则扩展名为“wav”。

- [Y，FS，NBITS]＝wavread(FILE)返回抽样频率FS(Hz)和文件中用来编码数据样本点的位数(NBITS)。

wavwrite：写入.wav声音文件。

- wavwrite(Y，FS，NBITS，WAVEFILE)将数据Y写入Windows WAVE文件，文件名由WAVEFILE指定，FS为抽样频率(Hz)，NBITS为位数，可为8、16、24或32。立体声的数据需要使用列数为2的矩阵表示。

zplane：z平面零极点图。

- zplane(Z，P)绘制出以单位圆为参考的零点Z和极点P的图，Z、P都是列向量。图中零点用“o”表示，极点用“x”表示。多重零极点的重数标注在各零极点的右上方。zplane(B，A)函数中，B和A都是行向量，向量中各元素分别表示系统函数的分子多项式$B(z)$、分母多项式$A(z)$的系数。

部分习题答案

第 1 章

1.1 电路的环路方程为

$$v_i = v_c + v_o$$

并且 $v_o = iR$，

$$v_c = (1/C)\int i\mathrm{d}t = (1/RC)\int v_o \mathrm{d}t$$

则环路方程可以写为

$$(1/RC)\int v_o \mathrm{d}t = v_i - v_o$$

利用(1.8)中梯形积分法，我们得到

$$v_i(n) - v_o(n) = v_i(n-1) - v_o(n-1) + (T/2RC)[v_o(n) + v_o(n-1)]$$

化简得

$$v_o(n) = K_1[v_i(n) - v_i(n-1)] + K_2 v_o(n-1)$$

其中

$$K_1 = \frac{2RC}{2RC+T}$$

$$K_2 = \frac{2RC-T}{2RC+T}$$

1.3 (a)

n	v_i	$v_i/3$	$v_i(n-1)/3$	$v_o(n-1)/3$	v_o
0	1	1/3	0	0	1/3
1	0	0	1/3	1/9	4/9
2	0	0	0	4/27	4/27
3	0	0	0	4/81	4/81
4	0	0	0	4/243	4/243
…	…	…	…	…	…

(b)

n	v_i	$v_i/3$	$v_i(n-1)/3$	$v_o(n-1)/3$	v_o
0	1	1/3	0	0	1/3
1	1	1/3	1/3	1/9	7/9
2	1	1/3	1/3	7/27	25/27
3	1	1/3	1/3	25/81	79/81
4	1	1/3	1/3	79/243	241/243
…	…	…	…	…	…

1.5 ·建立一个正弦输出的函数发生器，其振幅为 1V。

·将函数发生器产生的正弦信号输入至数字滤波器，使用双踪示波器观察输入和输出。

·增益是输出的振幅除以输入的振幅，相位是以输入为参考的相移。测量从近似于 $0 \sim f_S/2$ 频率范围的增益和相位。

1.7 $h(nT) = \{b_0, b_1, b_2, b_3, b_4, b_5, 0, 0, 0, \cdots\}$

1.9 (a) 幅度响应为

$$\left|\frac{y\mathrm{e}^{\mathrm{j}\omega nT}}{x\mathrm{e}^{\mathrm{j}\omega nT}}\right| = \frac{1}{\sqrt{1-K\cos(\omega T)^2 + K^2\sin^2(\omega T)}}$$

相位响应为

$$\theta(\omega T)=0-\arctan\left[\frac{K\sin(\omega T)}{1-K\cos(\omega T)}\right]$$

(d) 当 $K=1.25$ 时系统不稳定。

1.11　$h(nT)=\{0.25, 0.25, 0.5, 0, 0, 0, \cdots\}$

1.13　(a) $|K|>1$

(b) $|K|<1$

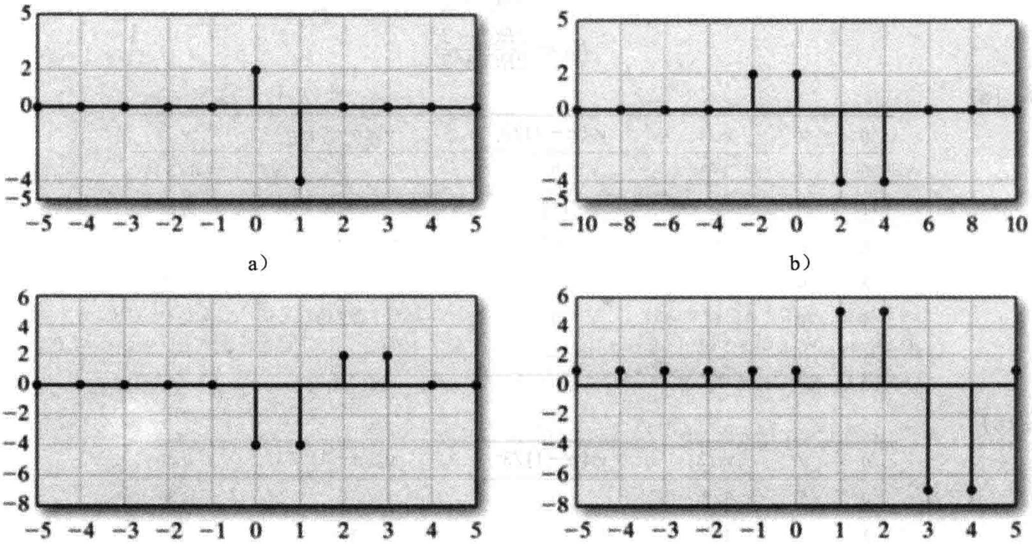
第 2 章

2.1　$x_d[n]$在 $n=2$ 时有非零值。

2.3　(a) $y[n]=y[n-1]+\frac{T}{2}(x[n]+x[n-1])$

(b)
```
y(1)=0;
T=0.1;
for n=1:51
  y(n+1)=y(n)+T/2*(exp(-n*T)+exp(-(n-1)*T));
end
y
```
最后结果：$y(5)\approx y[51]=0.9941$。

2.5



a)　b)　f)　h)

2.9　$x[n]=\frac{1}{A}y\left[\frac{1}{a}n-\frac{b}{a}\right]-\frac{B}{A}$

2.11　(a) 都不是

(c) $x_e[n]=1.5u[n-2]+1.5u[-n-2]$，$x_o[n]=1.5u[n-2]-1.5u[-n-2]$

2.13　(a) $x_1[n]$是周期的；$N_0=10$，$k=1$。

$x_2[n]$是周期的；$N_0=16$，$k=1$。

(b) $N_0=80$

2.17　(a) $\tau=0.083$s。

(b) $\tau=0.144$s，$\omega=1200\pi$ rad/s

2.20　(a) $x[n]=(0.5)^n u[n]$

2.21　(a) 线性　(b) 时不变　(c) 因果　(d) 不稳定

2.23　(a) 因果　(b) 非因果

2.25 $y[n]+0.7y[n-1]+0.1y[n-2]=x[n]+x[n-1]$

2.27 (a) $h[n]=h_1[n]*h_2[n]=n(0.8)^n u[n]$

2.29 $y[n]=0,\ n<0,\ y[0]=2,\ y[1]=3,\ y[2]=1,\ y[n]=0,\ n\geqslant 3$

第 3 章

3.1 (a) (i) $C_0=3,\ C_1=5/2,\ C_2=3\angle-\pi/4,\ C_k=0,\ k>2$

(b) (i) $C_0=3,\ 2|C_1|=5,\ \theta_1=0°,\ 2|C_2|=6,\ \theta_2=\pi/4,\ 2|C_k|=0,\ k>2$

3.3 (a) $m\neq n$

3.4 (a) $C_k=\frac{\mathrm{j}3}{k\pi}[1-\cos(k\pi/2)]$

(b) $C_k=\frac{-1}{k^2\pi^2}[1-(1+\mathrm{j}k\pi)\mathrm{e}^{-\mathrm{j}k\pi}]$

3.7
```
fs=5000;
pts=1000;
terms=3;
f0=10
y=3*fouriersq(fs,pts,f0,terms);
y(1001)=0
dt=1/fs;
t=(0:dt:pts*dt);
plot(t,y), grid
```

3.8 (a) $X_a(\omega)=-12\mathrm{sinc}(3\omega/2)$

(b) $X_b(\omega)=12\mathrm{e}^{-\mathrm{j}3w}\mathrm{sinc}(3\omega)$

3.11 (a) $G_4(\omega)=0.1\mathrm{sinc}(0.05\omega)+0.2\mathrm{sinc}(0.1\omega)$

$G_6(\omega)=0.05\mathrm{sinc}(0.05\omega)+0.2\mathrm{sinc}(0.01\omega)$

3.13 (a) $G_p(\omega)=\sum_{k=-\infty}^{\infty}2\pi\mathrm{sinc}\left(\frac{n\pi}{2}\right)\delta\left(\omega-\frac{n\pi}{2}\right)$

3.15 (a) $Y_a(\omega)=\pi^2[u(\omega+1)-u(\omega-1)]$

3.17 图 P3.12(a)电路中输入为

$$v_1(t) = 2\sin(50t)$$
$$V_2(t) = 1.99\sin(50t-78.7°)\mathrm{V}$$

3.19 (a) $X(\Omega)=\frac{1}{1-0.5\mathrm{e}^{-\mathrm{j}\Omega}}$

3.21 (a) $h[n]=2\delta[n]+3\delta[n-1]+4\delta[n-2]$

(b) $H[\Omega]=2+3\mathrm{e}^{-\mathrm{j}\Omega}+4\mathrm{e}^{-\mathrm{j}2\Omega}$

3.23 $X(\Omega)=\frac{2\pi}{5}\sum_{k=-\infty}^{\infty}(1+\mathrm{e}^{-\mathrm{j}2\pi k/5}+\mathrm{e}^{-\mathrm{j}4\pi k/5})\delta(\Omega-2\pi k/5)$

3.25 $X[k]=[1.9922,\ 1.1861-\mathrm{j}0.6487,\ 0.7969-\mathrm{j}0.3984,\ 0.6889-\mathrm{j}0.1799,\ 0.6641,\ 0.6889+\mathrm{j}0.1799,\ 0.7969+\mathrm{j}0.3984,\ 1.1861+\mathrm{j}0.6487]$

3.29 (b) $\Delta f=0.15625\mathrm{Hz}$

3.31 (a) $X[k]=[2.5000, 2.6429+\mathrm{j}0.8057, 3.4549+\mathrm{j}2.1353, 15.4409+\mathrm{j}11.9860, -5.5902, 15.4409-\mathrm{j}11.9860, 3.4549-\mathrm{j}2.1353, 2.6492-\mathrm{j}0.8057]$

(b)
```
T=0.1;
t=[0:T:7*T];
x=5*cos(8*pi.*t);
X=fft(x,8)
figure(1), stem(w,mX)
Xs=fftshift(X);
mX=abs(Xs);
w=[0:pi/(4*T):7*pi/(4*T)];
figure(1), stem(w,mX)
```

(c) $X(\omega)=5\pi[\delta(\omega-8\pi)+\delta(\omega+8\pi)]$；考虑 FFT 过程中的频谱扩散。

3.33 $X_{\text{Hann}}[k]=[0.1015, 0.1067-\text{j}0.0448, -4.0277-\text{j}0.8262, 7.5828+\text{j}3.3671, -7.4252, 7.5828-\text{j}3.3671, -4.0277+\text{j}0.8262, 0.1067+\text{j}0.0448]$

两题 FFT 的幅度频谱图表明，当使用汉宁窗时会有更少的频谱扩散。

3.34 (a) $X[k]=[6,\ -1-\text{j}1,\ 0,\ -1+\text{j}1]$

(b)
```
x= [1 2 2 1];
X= fft(x)
```

3.36 (a) $x[n]\otimes y[n]=[1,\ 3,\ -2,\ 5]$

(b) $x[n]*y[n]=[2,\ 5,\ 4,\ 5,\ -1,\ -2,\ -6]$

3.41 (c) $X(s)=\frac{3}{s}(\mathrm{e}^{-s}-\mathrm{e}^{-2s})$

3.42 (i) (a) $H(s)=\frac{2}{s^2+5s+4}$

(b) $h(t)=\frac{2}{3}(\mathrm{e}^{-t}-\mathrm{e}^{-4t})u(t)$

(c) $s(t)=\left(\frac{1}{2}-\frac{2\mathrm{e}^{-t}}{3}+\frac{1\mathrm{e}^{-4t}}{6}\right)u(t)$

3.44 (a) $|H(\omega)|=\frac{20}{\sqrt{\omega^4+\omega^2}}$，$\angle H(\omega)=\arctan\left(\frac{1}{-\omega}\right)$

3.45 (a)
```
numH = 2;  denH = [1  5  4];
[H,w] = freqs(numH,denH);
mH = abs(H); phH = angle(H)*180/pi;
figure(1), plot(w,mH), grid
figure(2), plot(w, phH), grid
```

3.47 (a) $|a|<|b^2|$

(b) $\frac{z}{z-a}+\frac{z}{z-b^2}$，$|a|<|z|<b^2$

3.51 (c) $X(z)=\frac{(0.5z^{-1})^{-10}-(0.5z^{-1})^{-21}}{1-(0.5z^{-1})}$，$z\neq 0$

3.53 (i) $x[n]=-1.5u[-n-1]+1.5(0.6)^n u[-n-1]$

(ii) $x[n]=-1.5u[n]-1.5(0.6)^n u[n]$

(iii) $x[n]=-1.5u[-n-1]-1.5(0.6)^n u[n]$

第 4 章

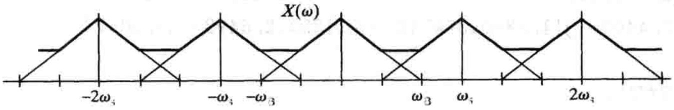
4.1 43100Hz，45100Hz，87200Hz。

4.3 (a) kf_S (b) f_S/k (c) f_S

4.5 $y_1(nT)=\sin(1800\pi nT)$并且 $y_2(nT)=\sin(2200\pi nT)$。

4.7 200Hz 的正弦信号，每个周期 5 个抽样点。

4.9 黑色曲线表示由于频谱重叠而导致的频率响应。



4.11 QSNR 为 52.64。

4.13 方波的 QSNR 为 64.983396。

三角波的 QSNR 为 53.684123。

4.15 (a) 10.5μs

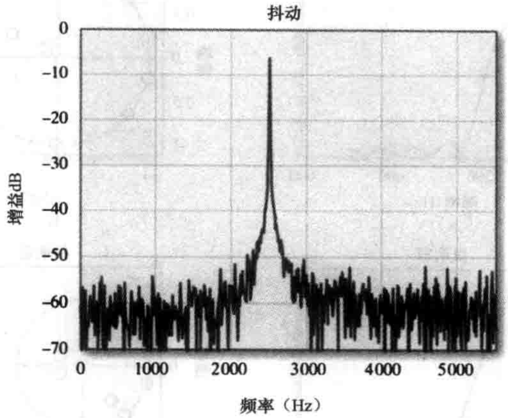
(b) 8.4μs

4.17 −49.89dB

4.19 $b=8$

4.21 $\approx R_{pa}+R_{pd}$

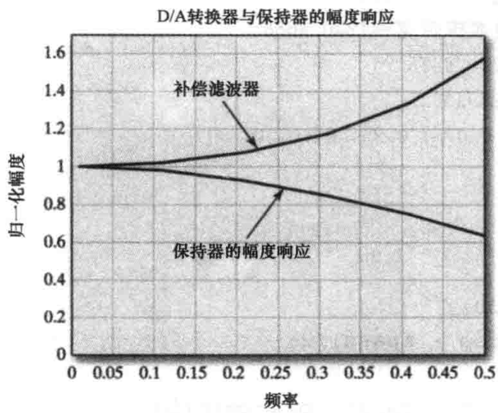
4.23 取决于 MATLAB 中的随机数发生器，产生的结果将会有所不同。



4.25 差分方程为：$y(n)=y(n-1)+(T/2)[x(n)+x(n-1)]$

4.27 现考虑一个 256V 的系统，则 8 位输出时误差为 0.5V。对于 4 位 D/A 转换器可以输出 16 个电平，在 256V 系统中输出误差为 8V。若利用 4 位 D/A 转换器模拟减法器，要求其误差为 0.5V，则 4 位 D/A 转换器的精度应提高 16 倍，误差为 0.5V 而不是 8V。

4.29



4.31 0.0075

第5章

5.1 (a) FIR，稳定，非因果，混合相位，HP。

(c) FIR，稳定，因果，混合相位，LP。

(e) FIR，稳定，因果，混合相位，HP。

5.3 (a) $h(nT)=\{0,0,0,0,0,0,0,0,0,1,0,0,\cdots\}$

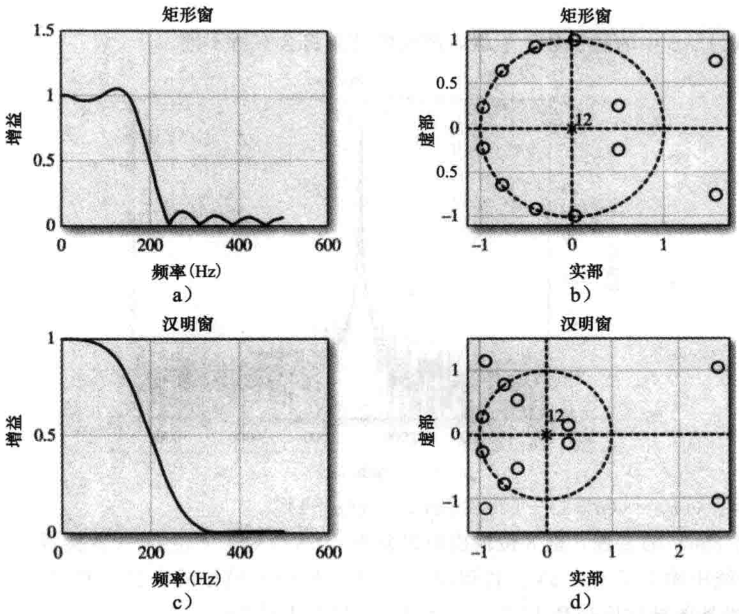
(c) 是，但是最小相位也是最大相位。严格地说，因为没有零点，所以最小相位和最大相位在此不适用。

5.5 $H(z)=\dfrac{0.2z^4+0.533z^3+z^2+0.533z+0.2}{z^4}$

5.7 $h=[0.0275\quad -0.0345\quad -0.0773\quad -0.0286\quad 0.1187\quad 0.2842\quad 0.3565\quad 0.2842\quad 0.1187\quad -0.0286\quad -0.0773\quad -0.0345\quad 0.0275]$

5.9 $W[n]=\{\cdots,0,0,0,4,-6,2,-6,4,0,0,0,\cdots\}$

5.11



5.13 矩形窗过渡带带宽＝556.6406

布莱克曼窗过渡带带宽＝2490.2344

三角窗(巴特利特窗)过渡带带宽＝1230.4688

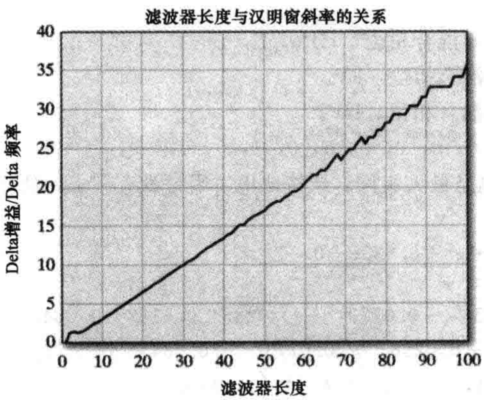
汉明窗过渡带带宽＝1210.9375

汉宁窗过渡带带宽＝1289.0625

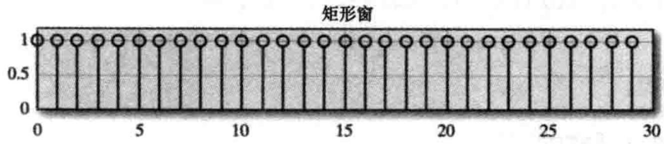
5.15 $h(n)=\{0, 0.5, 0.5, 0.075, 0.25, 0, 0, 0, \cdots\}$

5.17
```
fs = 44100;
fpass = 6000;
fstop = 8500;
rpass = .01;
rstop = 10^(-70/20);
fc = fpass + (fstop - fpass)/2;
len = 86;
num = fir1(len-1, fc/(fs/2), blackman(len));
```

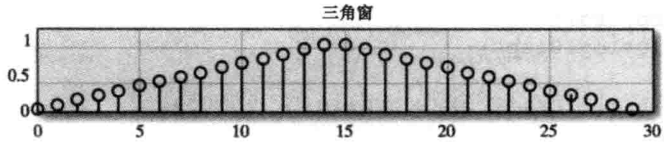
5.19



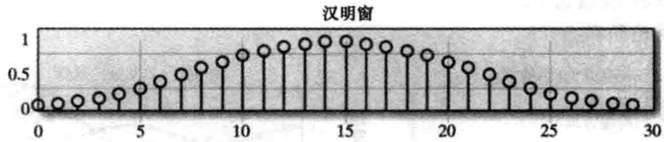
5.21



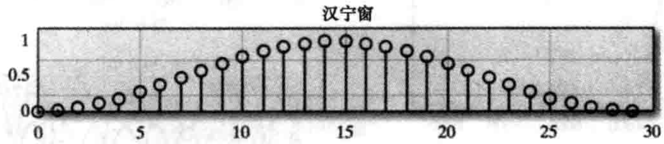
a)



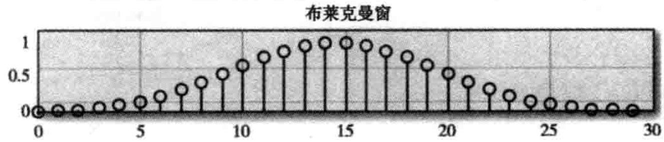
b)



c)



d)



e)

5.23
```
fs = 44100;
fpass = 8000;
Rp = 0.01;
fstop = 10000;
Rs = .02;
%Parks-McClelland
F = ([fpass fstop]);
M = ([1 0]);
Dev = [Rp Rs];
[N F A W] = firpmord(F, M, Dev, fs);
disp(N);
N = 40;
num = firpm(N, F, A, W);
%Kaiser
fc = fpass + (fstop-fpass)/2;
eps = min(Rp, Rs);
A = -20*log10(eps);
Ka = (A - 7.95)/14.36;
alpha = 0.1102*(A - 8.7);
L = ceil(1 + Ka*fs/(fstop - fpass));
disp(L-1);
```

```
L = 52;
num = fir1(L-1, fc/(fs/2), kaiser(L, alpha));
```

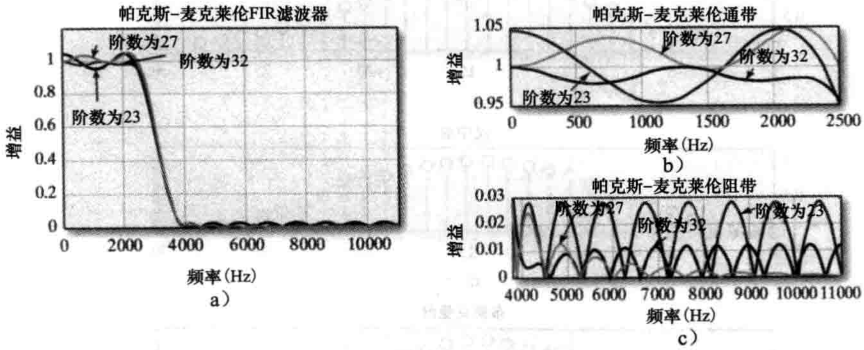
5.25

```
fs = 44100;
fpass = 3000;
fstop = 4600;
fc = (fpass + fstop)/2;
rp = .05;rs = .03;
eps = min(rp, rs);
lambda = -20*log10(eps);
L = 52;
num = fir1(L-1, fc/(fs/2), chebwin(L, lambda));
```

5.27 凯瑟阶数为27。

Dolph-Chebyshev 阶数为32。

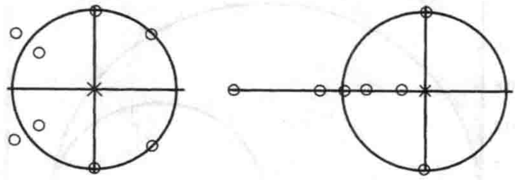
帕克斯-麦克莱伦阶数为23。



5.29

```
fs = 44100;T = 1/fs;
pts = 2500;
f0 = 500;
terms = 18;
x = fouriersq(fs, pts, f0, terms);
fpass = 5*f0;
fstop = 9*f0;
rpass = 0.05;
rstop = 0.05;
F = [fpass fstop];
M = ([1 0]);
Dev = [rpass rstop];
[N F A W] = firpmord(F, M, Dev, fs);
disp(N);
N = 28;
num = firpm(N, F, A, W);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
y = filter(num, 1, x);
figure(1);
subplot(2, 1, 2);
plot(nT, y, 'r');
axis([0 5/f0 -1.2 1.2]);
title('Filtered square wave');
xlabel('Time in seconds');
ylabel('voltage');
```

5.31



a)　　　　b)

5.33 $H_1(z)=\frac{z^2+1.2135z+0.5625}{z^2}$有最小相位。

5.35 (a) 零

(b) 乘以传输函数$(z-1)/z$。

5.37 希尔伯特变换器是20阶。

5.39 问题中增加下面代码，

```
n = 0:2047;
figure(1);clf;
subplot(2,1,1)
plot(n,x)
axis([0 2000 0 2])
title('Original signal');
fs = 1000;
diff = remez(21, [0 1], [ 0 pi/fs], 'd');
y = filter(diff, 1, x);
y = y/max(y);
subplot(2,1,2)
plot(n,y)
title('Differentiated signal');
axis([0 2000 0 2])
```

5.41 $h(n)=\{0.2, 0.2, 0.2, 0.02, 0.02, 0, 0, 0, \cdots\}$

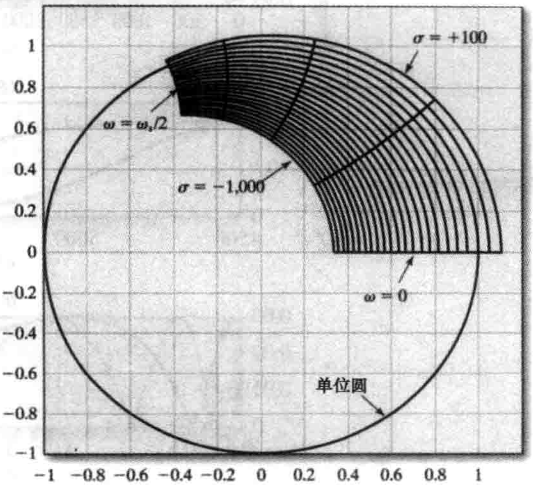
第6章

6.1 (a) $\frac{Y(z)}{X(z)}=\frac{z^2+4z+1}{z^2+1}\Rightarrow\frac{1}{s}$

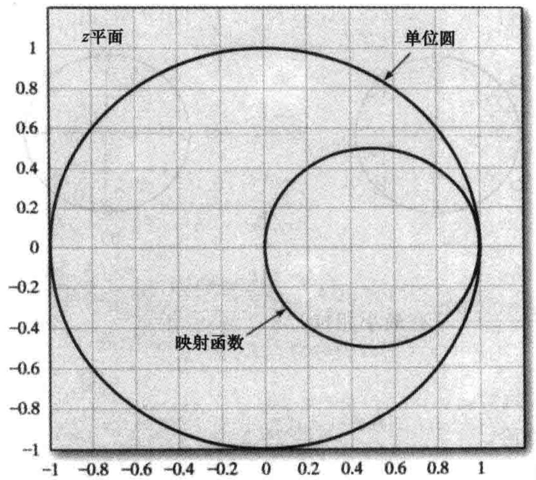
(b) 增加模拟滤波器的阶数一倍。

6.3 $f_{平面}=0, 96.896, 211.922, 286.045, 319.546$

6.5



6.7



模拟滤波器在 s 域稳定，则其对应的数字滤波器在 z 域也稳定。

6.9 $\omega_{\text{prewarp}} \underset{t\to 0}{\Rightarrow} \omega_c$

6.11 (a) 1。(b) 不是。(c) 分子为1阶，分母为2阶。(d) $f=8.926\text{Hz}$

6.13 (a) 当 K_1 为有限值时，稳定。

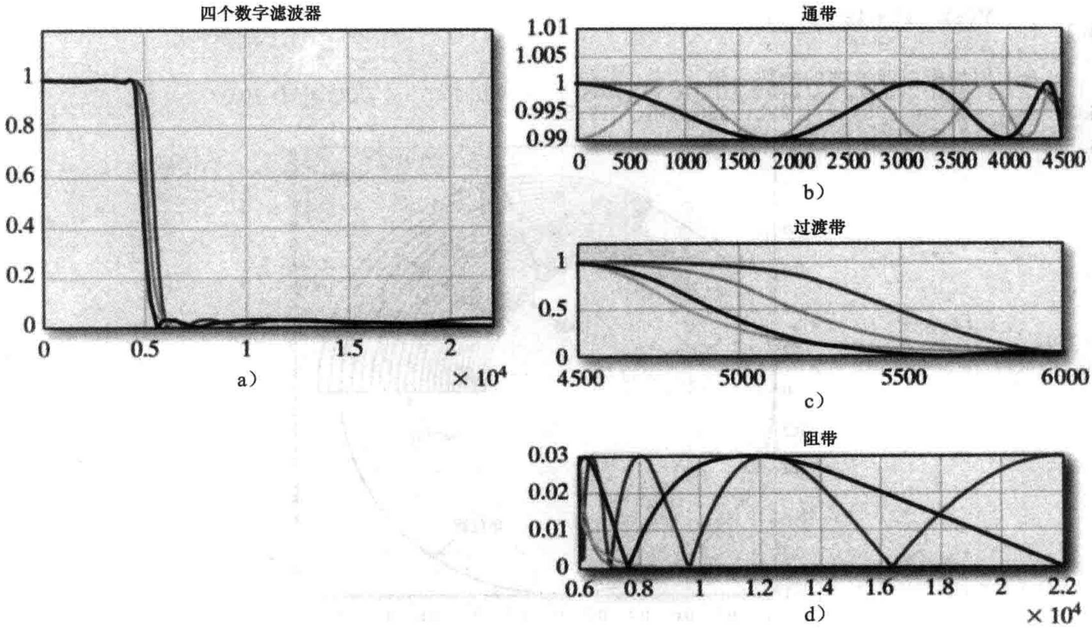
(b) 当 $|K_2|<1$ 时，稳定。

(c) 当 $|K_2|<1$ 时，稳定。

6.15
```
fs = 1000;T = 1/fs;
fL = 150;wL = 2*pi*fL;
fH = 150;wH = 2*pi*fH;
K = 0.1275;
numL = K*[1 2 1];
denL = [1 -0.7818 0.2919];
```

6.17 只有(a)和(c)。

6.19



a) b) c) d)

Butterworth——蓝色，Chebyshev Ⅰ型——红色，Chebyshev Ⅱ型——绿色，椭圆——黑色。输出结果为

```
Butterworth order is 18
Chebyshev 1 order is 8
Chebyshev 2 order is 8
elliptic order is 5
```

6.21
```
fs = 1000;
fpass = 150;
Rpass = 0.03;RpDB = -20*log10(1-Rpass);
fstop = 200;
Rstop = 0.03;RsDB = -20*log10(Rstop);
[N fp] = ellipord(fpass/(fs/2), fstop/(fs/2), RpDB, RsDB);
[num den] = ellip(N, RpDB, RsDB, fp);
[H f] = freqz(num, den, 1024, fs);
```

6.23 增益为0.0113。

6.25 增益为0.6169。

6.27 $N=20$

6.29 chirp函数的时域响应包络是幅度谱。

6.31 (a) $z=+1$

(b) $z=-1$

(c) $z=j$

6.33 $K=-0.2949$。

6.35 (a) 180°

(b) 滤波器是稳定的，所以脉冲响应一定会衰减至零。

(c) 输出为−7.5。

6.37 (a) 高通滤波器

(b) 阶数为2。

(c) 是。

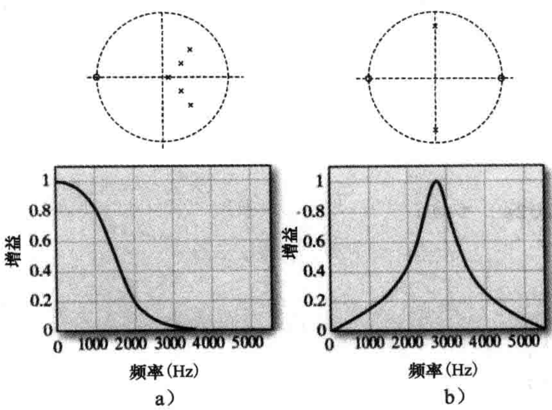
(d) $y(n)=x(n-1)-x(n-2)-1.2y(n-1)-0.52y(n-2)$

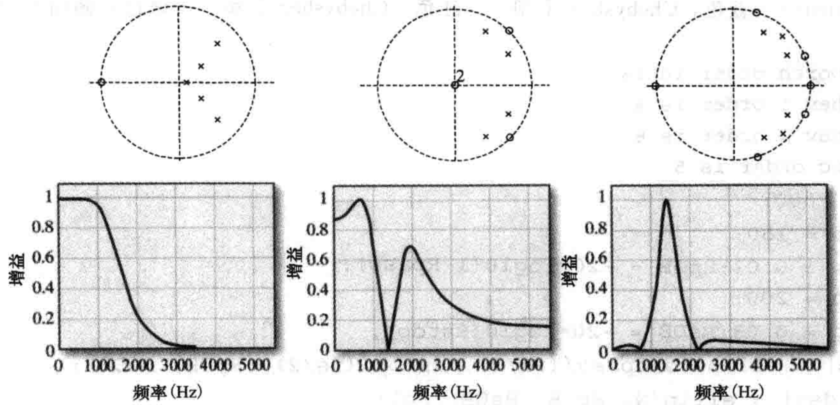
(e) 衰减到0。

(f) $h(n)=\{0, 1, -2.2, \cdots\}$

6.39 53.94dB

6.41





d)　　e)　　f)

6.43 $H_1(z) = \dfrac{-z^2 + 1.2z - 0.610}{z^2}$

$$H_2(z) = \frac{0.61z^2 - 1.2z + 1}{z^2}$$

$$H_3(z) = \frac{-0.61z^2 + 1.2z - 1}{z^2}$$

6.45 一种可行的解法

```
fs = 11025;
fpass = 2000; Rpass = .05;
fstop = 2800; Rstop = .05;
p1 = .79*exp(j*pi*.42);
p2 = conj(p1);
p3 = 0.5*exp(j*pi*.40);
p4 = conj(p3);
p5 = 0.2*exp(j*pi*.45);
p6 = conj(p5);
Ps = [p1 p2 p3 p4 p5 p6 ];
%
z1 = exp(j*pi*.75);
z2 = conj(z1);
z3 = exp(j*pi*.52);
z4 = conj(z3);
z5 = -1;
Zs = [z1 z2 z3 z4 z5];
den = poly(Ps);
num = poly(Zs);
[H f] = freqz(num, den, 1024, fs);
figure(1);clf
subplot(1, 2, 1);
zplane(num, den);
subplot(1, 2, 2);
plot(f, abs(H)/abs(H(1)));
xlabel('frequency in Hz');
ylabel('Gain');
line([0 fpass], [1-Rpass 1-Rpass]);
line([0 fpass], [1+Rpass 1+Rpass]);
line([fstop fs/2], [Rstop Rstop]);
```

6.47
```
fs = 11025;
himp = [0.05874 0.24544 0.40875...
        0.32218 0.06298 -0.10723 ...
       -0.08514 0.02305 0.07536 ...
        0.03961];
h0 = himp(1);h1 = himp(2);h2 = himp(3);
h3 = himp(4);h4 = himp(5);h5 = himp(6);
h6 = himp(7);
%Form the A matrix
A = [1 0 0 0 0 0 0; ...
     0 1 0 0 -h0 0 0; ...
     0 0 1 0 -h1 -h0 0; ...
     0 0 0 1 -h2 -h1 -h0; ...
     0 0 0 0 -h3 -h2 -h1; ...
     0 0 0 0 -h4 -h3 -h2; ...
     0 0 0 0 -h5 -h4 -h3; ...
    ];
%Form the b matrix
b = [h0; h1; h2; h3; h4; h5; h6;];
%Ax = b so
x = A^-1*b;
%Get numerator and denominator from x
num = [x(1) x(2) x(3) x(4)];
den = [1 x(5) x(6) x(7)];
```

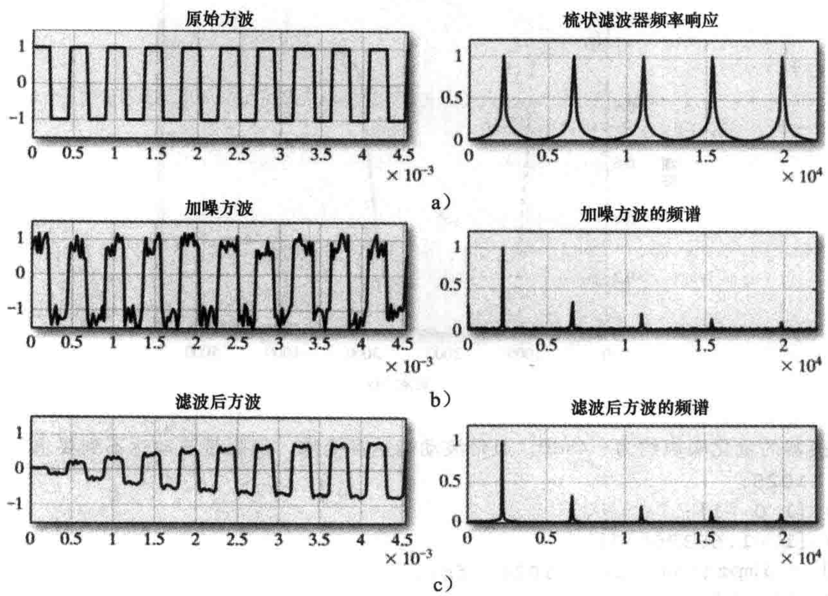
6.49 (a) $M=2$，$N=3$

(b) $h_I(n)=\{-1, -0.7, -0.1, 0.2, 0.1, 0.05\}$

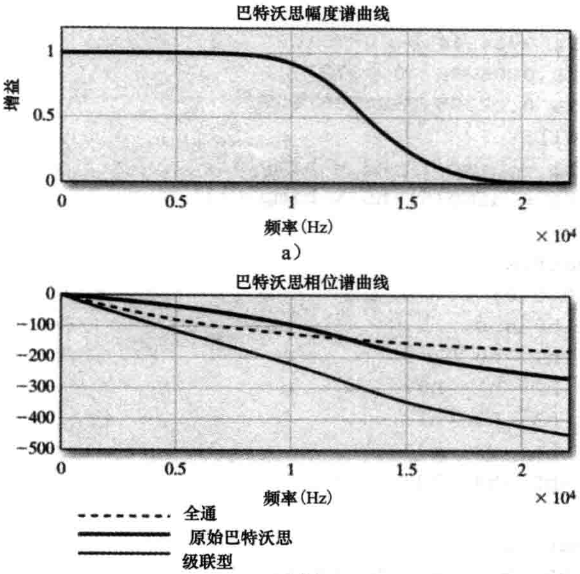
(c) $h_a(n)=\{-1, -0.7, -0.1, 0.2, 0.1, 0.05\}$

6.51 衰减$=-10\lg(V_o/V_i)^2$。在半功率点时，$(V_o/V_i)^2=1/2$，衰减$=-10\lg(1/2)=3.0103\approx3\text{dB}$。

6.53



6.55



a)

b)

6.57 单级中频带相位偏转约为−10°。二级中频带相位偏转约为−20°。幅度增益在整个频带内不变。

6.59 159.155Hz

6.61 (a) 使用线性内插法得到约−15.1dB。MATLAB 图给出更精确的结果为−12.45dB。

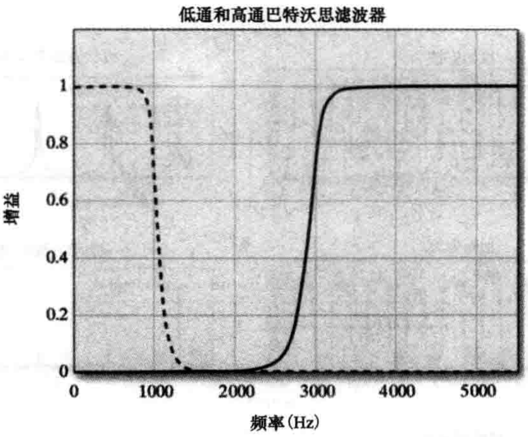
(b) $H(z)=K\dfrac{z^2-1.8596z+1}{z^2-1.8038z+0.9409}$

6.63 BW=0.50Hz，f_c=1333.3Hz。

6.65 极点位于 $r\cos(\theta_c)\pm jr\sin(\theta_c)=0.65512\pm j0.72758$ 处。

零点位于共振点 $z=\pm1$ 处。

6.67



6.69 8 位转换器的量化噪声约为−48dB。通带波动能达到要求，但阻带波动达不到要求。

6.71
```
fs = 11025;
num = [1 0 -1];
den = [1 -1.683906 1];
[h nT] = impz(num, den, 1024, fs);
figure(1);clf;
stem(nT, h);
title('Impulse Response');
xlabel('time in seconds');
```

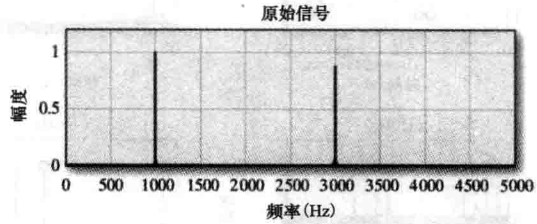
```
ylabel('voltage');
axis([0 0.01 -2.2 2.2]);
```

6.73 逆滤波器为$\frac{0.5z^2-0.7320z+0.4094}{z^2+0.657z}$

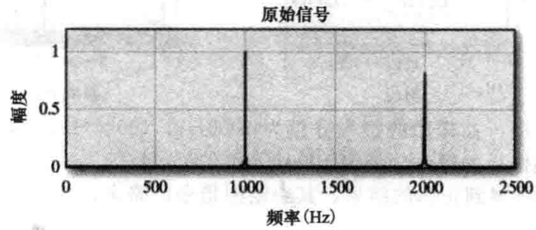
6.75
```
fps = 100;Rps = 0.01;
RpsDB = -20*log10(1-Rps);
fst = 150;Rst = 0.01;
RstDB = -20*log10(Rst);
[N fp] = ellipord(fps/(fs/2), fst/(fs/2),...
RpsDB, RstDB);
disp(N);
[num den] = ellip(N, RpsDB, RstDB, fp);
```

第7章

7.1



a)



b)

7.3 降抽样在5倍时会产生混叠。

7.5 $R_{pass}=0.02$

$R_{stop}=0.0001$

7.7 三个滤波器的通带都是从0Hz到100Hz。对于第一个滤波器，当$D=2$时，阻带从10kHz到20kHz。经过降抽样，新的抽样频率为20kHz。对于第二个滤波器，当$D=4$时，阻带从2.5kHz到10kHz。降抽样后，新的抽样频率为5kHz。对于最后一个滤波器，当$D=5$时，阻带从500Hz到2.5kHz。

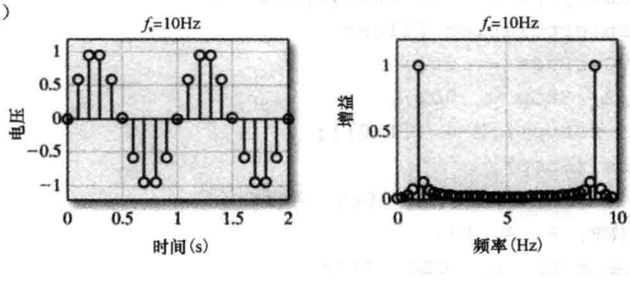
$$R_{pass}=0.001$$
$$R_{stop}=0.002$$

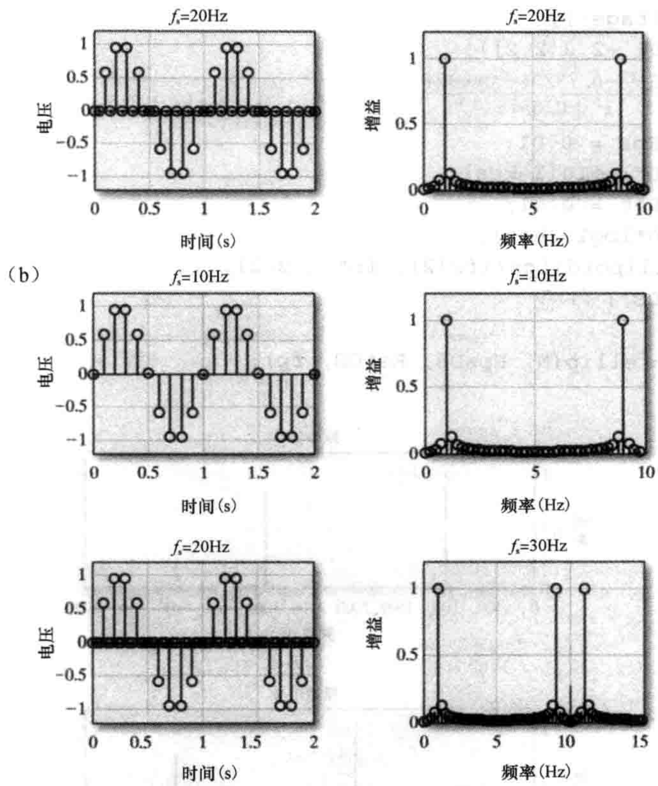
7.9 $(1-r_{pfinal})=(1-r_p)^3=1-3r_p+3r_p^2-r_p^3$

其中$r_{pfinal}\approx 3r_p$

7.11

(a)





7.13　第一个信号用 11025 进行抽样后的频率分量为 1000Hz、10025Hz、12025Hz 和 21050Hz。第二个信号用 22050 进行抽样后的频率分量为 1000Hz 和 21050Hz。

7.15　下列 MATLAB 程序将得到正确的结果。其中绘图指令已略去。

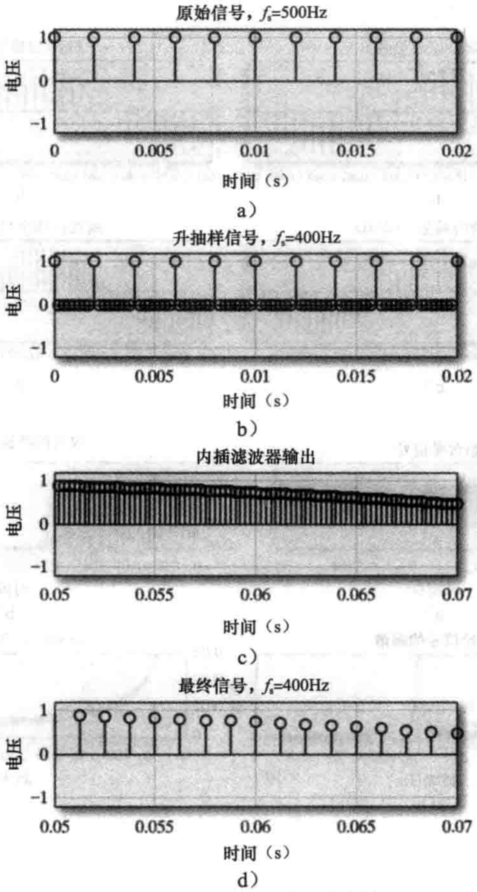
```
N = 10000;
fs = 11025;T = 1/fs;
t = 0:T:(N-1)*T;
y = chirp(t, 10, (N-1)*T, 2000);
%
yDFT = fft(y);
f = 0:fs/N:fs-fs/N;
yDFTNorm = abs(yDFT)/max(abs(yDFT));
%Upsample by 4
yUp = upsample(y, 4);
nTUp = [0:4*N-1]*T/4;
yUpDFT = fft(yUp);
f = 0:fs/(N):4*fs-fs/(4*N);
yUpNorm = abs(yUpDFT)/max(abs(yUpDFT));
% %Create interpolation filter
fpass = 2000;rPass = .001;
fstop = fs/2;rStop = .001;
F = ([fpass fstop]);M = ([1 0]);
Dev = [rPass rStop];
[NF F A W] = firpmord(F, M, Dev, 4*fs);
h1 = firpm(NF, F, A, W);
[H f1] = freqz(h1, 1, 1024, 4*fs);
```

```
%Filter result
yF = filter(h1, 1, yUp);
yF = yF/max(yF);
yFUpDFT = fft(yF);
f = 0:fs/(N):4*fs-fs/(4*N);
yFNorm = abs(yFUpDFT)/max(abs(yFUpDFT));
```

7.17 (a) 0.65625＝21/32。

(c) 2.375＝19/8。

7.19 8 倍升抽样，5 倍降抽样。



7.21 (a) 该程序为 2 倍降抽样，每隔一个丢弃一个样本值。每次模拟数字转换，计数变量都会递增。当计数变量为偶数时，标志位置 1，主程序做一次计算。当计数变量为奇数时，跳过该抽样点。

(b) 系数对称的 FIR 滤波器。并不是按照此式 y+＝b[i] * x[i]+b[i] * x[N−i]进行计算，我们提去公共因子 $b[i]$以减少乘法的次数。

(c) 计数变量通常和 redoFlag 标志位结合起来。当计数变量为偶数时，通过将标志位置 1 来进行一次计算。相反，当计数变量为奇数时，跳过该抽样点。

(d) 中断服务程序在每个样本抽样周期运行一次。

(e) 这是因为滤波器的系数具有对称性，$b[0]=b[13]$，$b[1]=b[12]$，…

(f) 可以采用相同的系统，但需要改变中断服务程序。原先是当计数变量 count 为偶数时，标志位置 1，启动中断服务程序。而当变为 3 倍降抽样时，需要检查计数变量 count 是否满足 count ％ 3为零，据此来设置标志位。

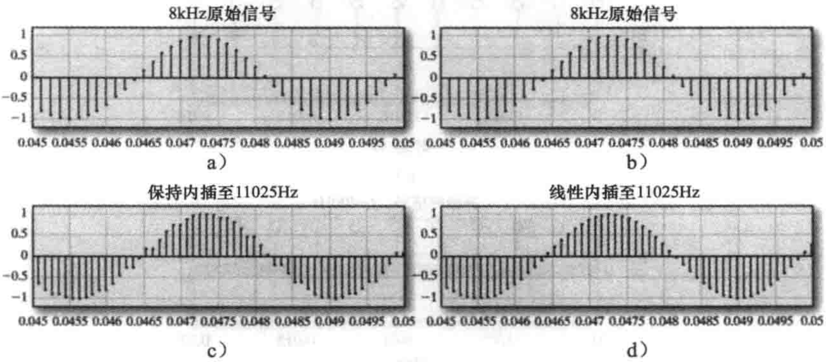
(g) 约有两个抽样周期。每个抽样周期都会有一个中断用来输入下一个样本。由于每隔一个的样

本将会被丢弃，所以有两倍的有效时间用于计算。

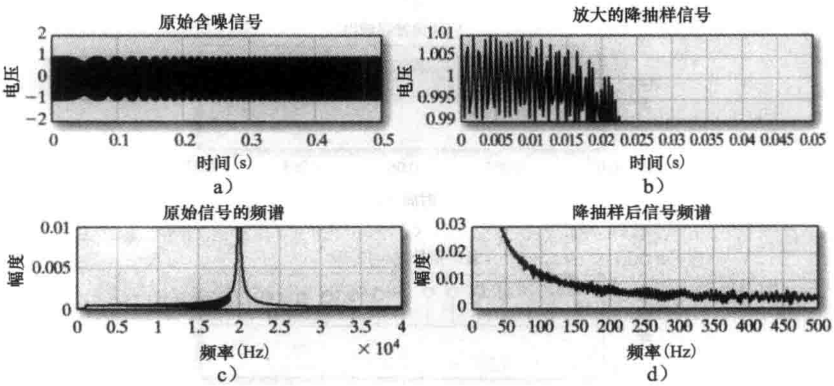
7.23 下列MATLAB程序可以提供正确的信号，输出命令已被省略。

```
fs = 1500;T = 1/fs;
fsig = 100;
n = 0:1/T;
nT = n*T;
x = chirp(nT, 10, 0.2, 500);
nT5 = n*T/5;
y5 = interp1(nT, x, nT5,...
'linear', 'extrap');
```

7.25



7.27



第8章

8.1 $(-44)+(-112)+64+125+(-44)=-11$

8.3 10位

8.5 $+1\sim3.98$V

$-1\sim6.02$V

8.7 对于两个二阶系统级联，需要7次乘法，8次加法。

对于PM运算法，需要10次乘法，17次加法。

8.9 $T=a+m(N+1)/2+s$ 且 $f_s\leqslant1/T$ 时忽略加法时间。

8.11 $\frac{z_d}{p_1}\frac{z_c}{p_2}\frac{z_b}{p_3}\frac{z_a}{p_4}$

8.13
```
fs = 8000;
fpass = 1000;
fstop = 2000;
rpass = 0.03;
rstop = 0.03;
```

```
F = ([fpass fstop]);
M = ([1 0]);
Dev = [rpass rstop];
[N F A W] = firpmord(F, M, Dev, fs);
N = 12;
disp(N);
num = firpm(N, F, A, W);
```

8.15
```
const float G = 0.00477006;
const float b[10][3] = {{1.0, -4.71106722, 5.28109714},
                        {1.0, -0.89206222, 0.18935459},
                        {1.0, -2.52279922, 2.76479423},
                        {1.0, 1.97588871, 1.00000000},
                        {1.0, 0.96627129, 1.00000000},
                        {1.0, 1.78732314, 1.00000000},
                        {1.0, 0.10550133, 1.00000000},
                        {1.0, 1.43418695, 1.00000000},
                        {1.0, 0.47004745, 1.00000000},
                        {1.0, -0.91247269, 0.36169057}
                       };

int main()
  {int i, xInt;
   float y;
   float x[10][3];
   while(1)
     {//Do A/D to xInt
      x[0][0] = ((float)xInt/1024.0;
      for(i=0;i<10;i++)
       {x[i+1][0] = b[i][0]*x[i][0] + b[i][1]*x[i][1] + b[i][2]*x[i][2];
       }
      y = G*x[9][0];
      //Send y to D/A
      for(i=0;i<10;i++)
       {x[i][2] = x[i][1];
        x[i][1] = x[i][0];
       }
     //Wait until next cycle
     }
  }
```

8.17

	3 位	4 位	6 位	12 位
0.5	100	1000	100000	100000000000
2.625	101	1011	101010	101010000000
15.015625	111	1111	111100	111100000100

8.19 (a) 抽样率越高，极点将越密集并且越接近 $z=+1$ 点。这将会增加系数的敏感度。

(b) 在 44100 处，有 $\Delta p_1=1.8431\Delta a_1$。

在 88200 处，有 $\Delta p_1=8.0641\Delta a_1$。

8.21 k1 = 0.25396 k2 = 0.41163 k3 = 0.65976

8.23 $\boldsymbol{A}\leftarrow\boldsymbol{T}^{-1}\boldsymbol{AT}$ 则 $\begin{bmatrix}0 & 1\\1 & 0\end{bmatrix}\begin{bmatrix}0 & 1\\-a_2 & -a_1\end{bmatrix}\begin{bmatrix}0 & 1\\1 & 0\end{bmatrix}=\begin{bmatrix}-a_2 & -a_1\\1 & 0\end{bmatrix}$

$\boldsymbol{B} \leftarrow \boldsymbol{T}^{-1}\boldsymbol{B}$ 则 $\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}\begin{bmatrix} 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$

$\boldsymbol{C} \leftarrow \boldsymbol{CT}$ 则 $[b_2 - b_0a_2 \quad b_1 - b_0a_1]\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} = [b_1 - b_0a_1 \quad b_2 - b_0a_2]$

$\boldsymbol{D} \leftarrow \boldsymbol{D}$ 则 $[1]$

8.25 $\boldsymbol{A1} = \begin{bmatrix} 0.4980 & -0.2466 \\ 0.2466 & 0.4980 \end{bmatrix}$ $\boldsymbol{B1} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$ $\boldsymbol{C1} = [-8.8544 \quad 2.9960]$ $\boldsymbol{D1} = [1]$

$\boldsymbol{A2} = \begin{bmatrix} 0.4687 & -0.6752 \\ 0.6752 & 0.4687 \end{bmatrix}$ $\boldsymbol{B2} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$ $\boldsymbol{C2} = [-2.5197 \quad 2.9374]$ $\boldsymbol{D2} = [1]$

8.27 这段 MATLAB 代码显示了如何实现将滤波器拆分成两级形式的过程，并且将每一级的增益输出。

```
sos = tf2sos(num, den);
% This loop pulls out the second order sections from sos
% and evaluates their magnitude response. The magnitude
% responses are multiplied to form the total response of
% the system.
num = sos(1:1,1:3); den = sos(1:1,4:6);
[h fd] = freqz(num,den,1024,fs);
h = abs(h);
for i = 2:N
  num = sos(i:i,1:3); den = sos(i:i,4:6);
  [h1 fd] = freqz(num,den,1024,fs);
  h = h.*abs(h1);
end
```

8.29 (a) $\boldsymbol{A} = \begin{bmatrix} 0 & 1 \\ -0.9 & -0.3 \end{bmatrix}$ $\boldsymbol{B} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$ $\boldsymbol{C} = [0.1 \quad 2.3]$ $\boldsymbol{D} = [1]$

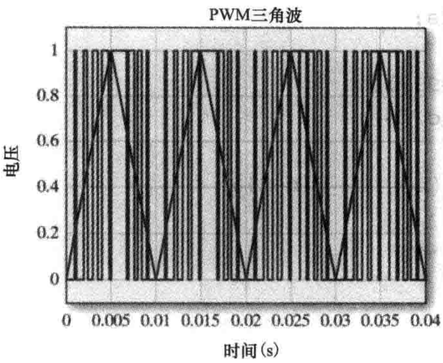
(c)
```
num = [b0 b1 b2];den = [a0 a1 a2];
r = roots(den);
alpha = real(r(1,1));beta = imag(r(1,1));
T = [-1/beta 0;-alpha/beta 1];
At = T^-1*A*T;
Bt = T^-1*B;
Ct = C*T;
```

8.31
```
1   const float b[] = {-0.00725966,  0.04014696,  0.02354041,
2                      -0.08357119, -0.02859982,  0.31030209,
3                       0.53138928,
4                       0.31030209, -0.02859982, -0.08357119,
5                       0.02354041,  0.04014696, -0.00725966
6                     };
7   int main()
8     {int i, xInt;
9      float x, y;
10     float v[13];
11     while(1)
12       {//Do A/D to xInt
13        x = (float)(xInt)/1024.0;
14        y = b[0]*x + v[0];
15        for(i=1;i<N-1;i++)
16          v[i-1] = b[i]*x + v[i];
17        v[N-1] = b[N]*x;
18        //Send y to D/A
```

```
19       //Wait until next cycle
        }
    }
```

第9章

9.1



9.3 对于无损音频文件，所有信息都进行编码，解码时文件会得到精确复原。有损音频编码时，文件中的部分信息会在编码时丢失。丢失的数据在原环境中通常都是多余的或者不易发现的。因此，有损编码比无损编码有更高的压缩比。

9.5 $\cos(x)=1-k_1x^2(1-k_2x^2(1-k_3x^2(1-\cdots$

其中，$k_1=1/(1\cdot 2)$，$k_2=1/(3\cdot 4)$，$k_3=1/(5\cdot 6)$，…

9.7 初始化 y，令其在单位1时为1。这是一个振荡器，需要一个小扰动来使其振动起来。

9.9 与100Hz混叠的最低频率为10925Hz。在表9.12中将

fsig=100; Tsig=1/fsig;

换成

fsig=10925; Tsig=1/fsig;

将获得相同结果。

9.11

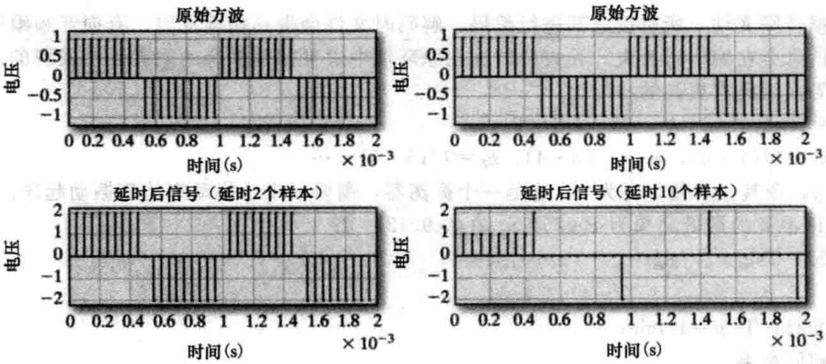
```
fs = 8000;Ts = 1/fs;
%fsig = 174.614;Tsig = 1/fsig;
fsig = 1318.51;Tsig = 1/fsig;
thetaIncr = Ts*2*pi/Tsig;
A = sin(thetaIncr);
B = cos(thetaIncr);
t = 0:Ts:5;
y1 = [1 zeros(1, length(t)-1)];
y2 = [1 zeros(1, length(t)-1)];
for n = 2:length(y1)
y1(n) = B*y1(n-1) + A*y2(n-1);
y2(n) = -A*y1(n-1) + B*y2(n-1);
end;
figure(1);clf;
y1 = y1/max(y1);
plot(t, y1);
holdon;
y2 = y2/max(y2);
plot(t, y2);
```

```
axis([0 .1 -1.5 1.5]);
xlabel('time in seconds');
ylabel('voltage');
title('sine and cosine at 0.393 Hz');
wavplay(y1, fs);
```

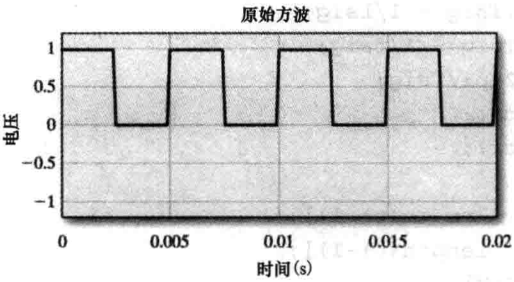
9.13

```
fs = 11025;T = 1/fs;
fo = 1000;
y = [1 zeros(1, 19999)];
alpha = cos(2*pi*fo/fs);
for n = 3:19999;
  y(n) = 2*alpha*y(n-1) - y(n-2);
end
kT = (1:length(y))*T;
figure(1);clf;
y = y/max(y);
plot(kT, y);
```

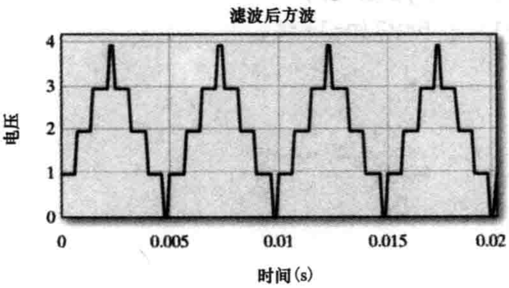
9.15



9.17



a)



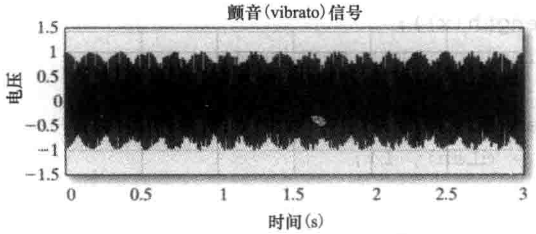
b)

9.19

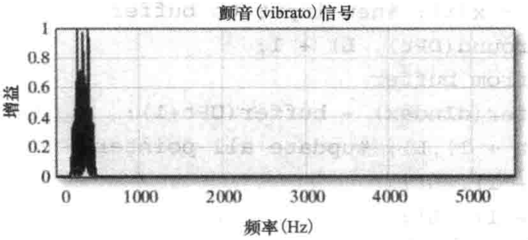
```
x = chirp(t, 20, 3, 2000);
N = fix(fs*dTimeMax);
L = N + 1; %buffer length
buffer = zeros(1, L);
y = zeros(1, length(x));
UPt = N; %Original pointer settings
P1 = 1;
dLen = fix(N*rand(1,1));
DPt = mod((UPt - dLen), L);
P2 = DPt;
j = 0;
for i = 1:length(x)
  buffer(UPt+1) = x(i); %new input to buffer
  dIndex = mod(round(DPt), L) + 1;
  %Take output from buffer
  y(i) = K1*buffer(dIndex) + buffer(UPt+1);
  UPt = mod((UPt + 1),L); %update all pointers
  P1 = mod((P1 + 1), L);
  P2 = mod((P2 + 1), L);
  DPt = mod((DPt + 1), L);
  %after 3000 samples DPt will equal P1 so reset
  % to a new random value
  if(j == 3000)
      dLen = fix(N*rand(1,1));
      P2 = P1;
      P1 = mod((UPt - dLen), L);
      j = 0;
  end
  j = j + 1;
  %Move DPt linearly closer to P2
  if(P2 > P1)
    DPt = DPt + ((P2 - P1)/3000);
  else
    DPt = DPt + ((P2 + L - P1)/3000);
  end
end
y = y/(max(y) - min(y)); %normalize
wavplay(y, fs); %play result
figure(1);clf;
subplot(2, 1, 2);
y = y/max(y);
plot(t, y);
axis([0 1 -1.2 1.2])
title('Filtered signal signal');
xlabel('time in seconds');
ylabel('voltage');
subplot(2, 1, 1);
x = x/max(x);
```

```
plot(t, x);
axis([0 1 -1.2 1.2]);
title('Original signal');
```

9.21



a)



b)

9.23 频率越低波长越长，从反射器反射的回声会与原信号产生干扰。然而，越高的频率波长越短，回声与原信号基本不会产生干扰。

9.25
```
fs = 11025; T = 1/fs;
t = T*[1:10000];
s1 = chirp(t, 480, 1, 500);
s2 = chirp(t, 500, 1, 480);
vol1 = (1:length(s1))/length(s1);
vol2 = (length(s2):-1:1)/length(s2);
s = [s1.*vol1 s2.*vol2];
t = T*[0:length(s)-1];
figure(1);clf;
plot(t, s);
title('frequency shift');
xlabel('time in seconds');
ylabel('voltage');
axis([0 2 -1 1]);
sound(s, fs);
```

9.27 (a) 若 $\omega_0=0$，$a_1=-2$，$b_1=-2/(1+k_1)$则

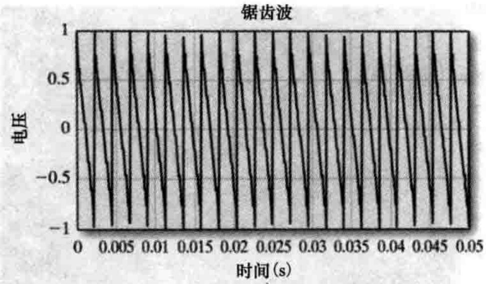
$$H(z)=\frac{1}{1+k_1}\,\frac{(1+Ak_1)z^2-2z+1-Ak_1}{z^2-2z/(1+k_1)+(1-k_1)/(1+k_1)}$$

$$H(z)=\frac{z^2-2z+1+zAk_1(z-1)}{z^2-2z+1+zk_1(z-1)}$$

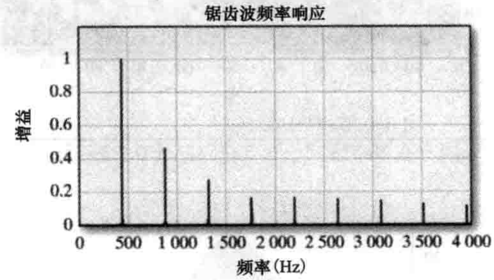
$$H(z)=\frac{(z-1)^2+zAk_1(z-1)}{(z-1)^2+zk_1(z-1)}=\frac{(z-1)+zAk_1}{(z-1)+zk_1}$$

$$H(z)=\frac{z(1+Ak_1)-1}{z(1+k_1)-1}$$

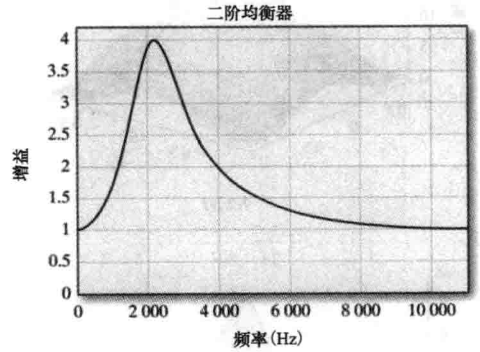
9.29



a)



b)



c)

第 10 章

10.1

	k_2: 0	1	2	3
k_1: 0	1	0.5	0	…
1	0.5	0.25	0	…
2	0	0	0	…
3	…	…	…	…

10.3

输出掩码

k_1					
0	0	0	0	0	k_2
0	0	0.5	1.0	0	
0	0	0.25	0.5	0	
0	0	0	0	0	
0	0	0	0	0	

输入掩码

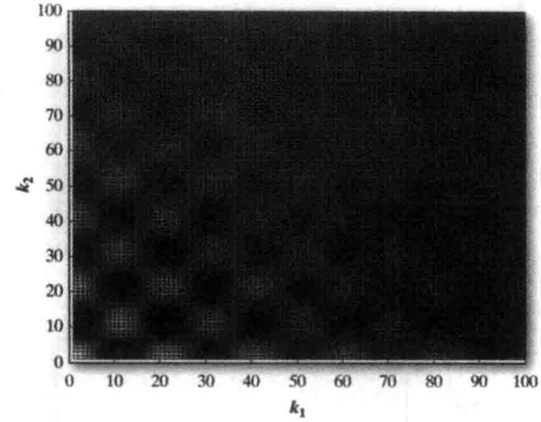
k_1					
0	0	0	0	0	k_2
0	0	0	1.0	0	
0	0	0	0	0	
0	0	0	0	0	
0	0	0	0	0	

10.5 $y(k_1,k_2)=-a_{0,1}y(k_1,k_2-1)-a_{1,0}y(k_1-1,k_2)-a_{1,1}y(k_1-1,k_2-1)$

$-a_{2,0}y(k_1-2,k_2)-a_{0,2}y(k_1,k_2-2)-a_{2,2}y(k_1-2,k_2-2)$

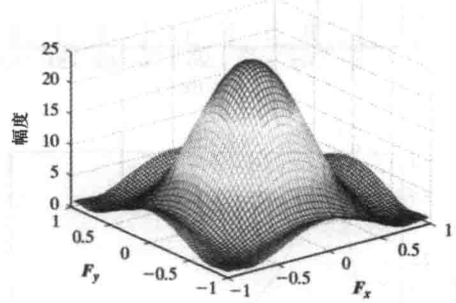
其中，$a_{0,1}=0.5$，$a_{1,0}=0.5$，$a_{1,1}=0.25$，$a_{2,0}=-0.1$，$a_{0,2}=-0.1$，$a_{2,2}=-0.05$

10.7

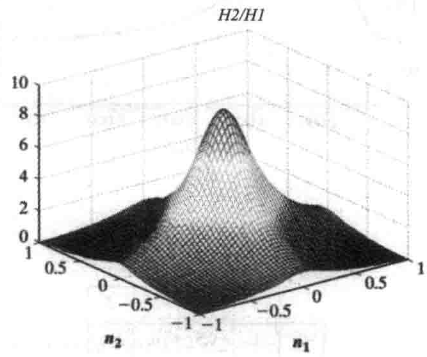


10.9　$H(z_1, z_2)=\dfrac{1+0.5z_2^{-1}+0.5z_1^{-1}}{1-0.25z_1^{-1}z_2^{-1}-0.5z_1^{-1}z_2^{-2}-0.5z_1^{-2}z_2^{-1}+0.25z_1^{-2}z_1^{-2}}$

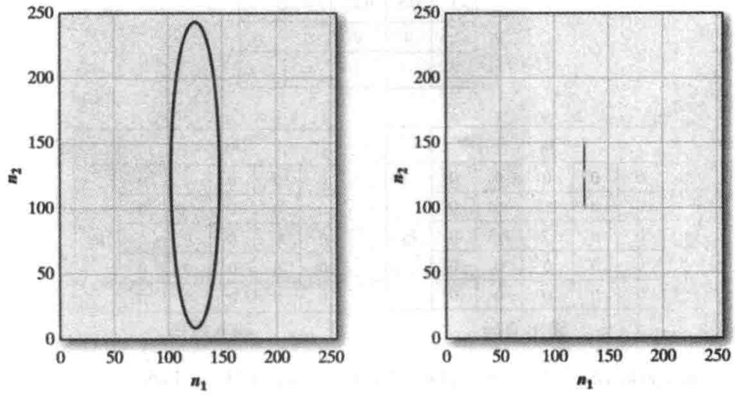
10.11



10.13　(a)



10.15



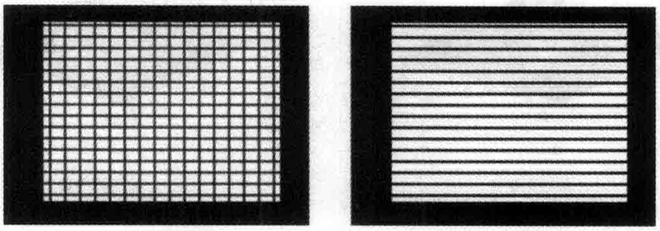
水平条纹　　线性调频脉冲

a)　　b)

10.17

```
1.0000  1.4000  0.7000       0       0
1.4000  2.9600  2.3800  0.7000       0
0.7000  2.3800  3.4500  2.3800  0.7000
     0  0.7000  2.3800  2.4500  0.9800
     0       0  0.7000  0.9800  0.4900
```

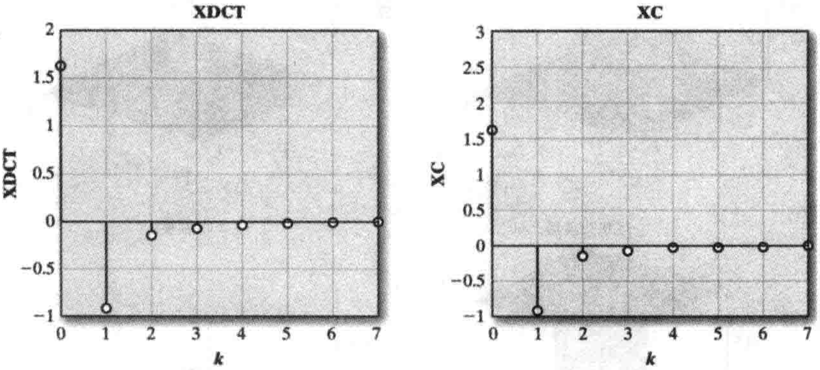
10.19



原始图像　　　FFT变换后的图像

a)　　　b)

10.21



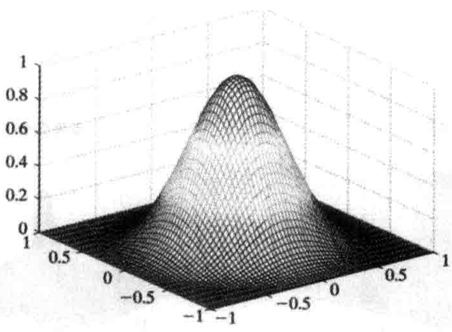
a)　　　b)

图 a 利用了 MATLAB 中 DCT 函数。图 b 利用 FFT 的结果。

10.23

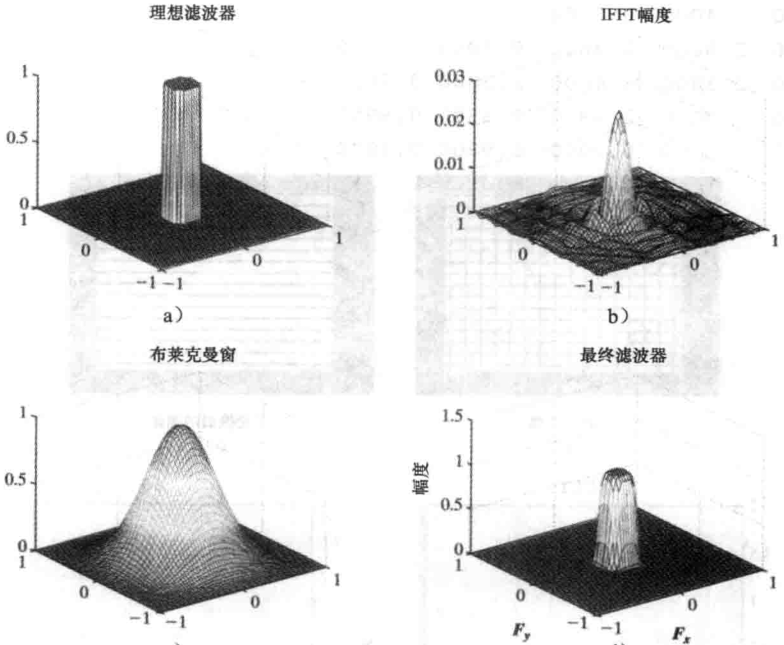
```
N = 65;
fc = .3;
[t1 t2] = freqspace(N, 'meshgrid');
r = sqrt(t1.^2 + t2.^2);
Hw = .42 +
.5*cos(pi*r)+.08*cos(2*pi*r);
Hw(r > 1) = 0;
figure(1);clf;
mesh(t1, t2, Hw);
```

a)



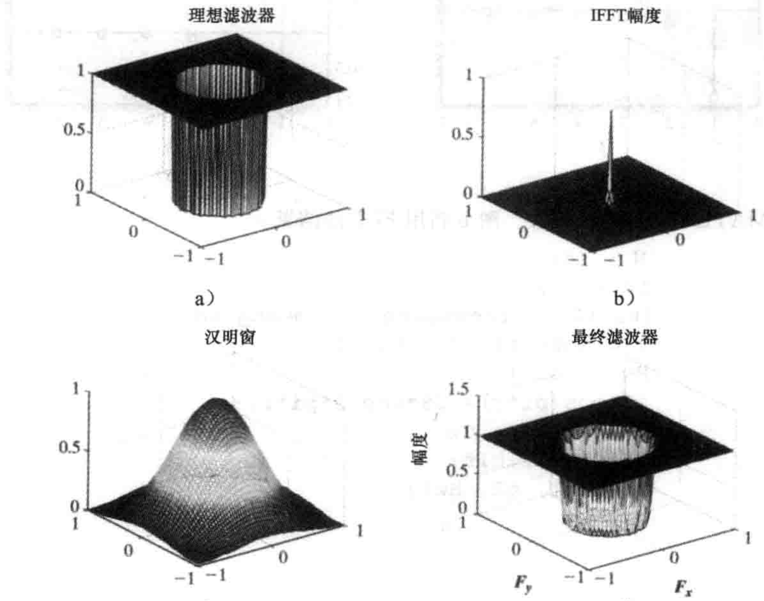
b)

10.25



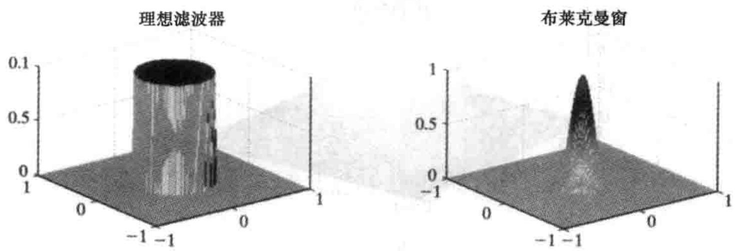
a）b）c）d）

10.27

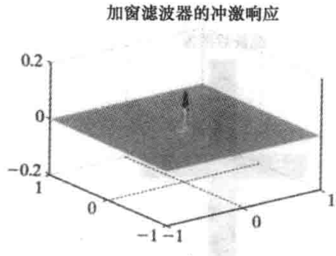


a）b）c）d）

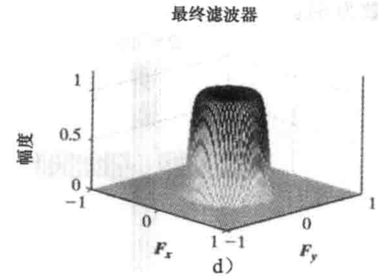
10.29



a）b）

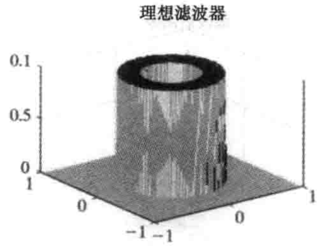


c）

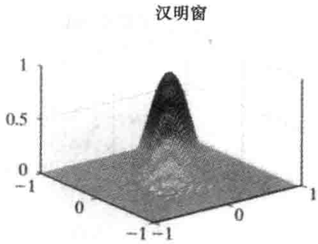


d）

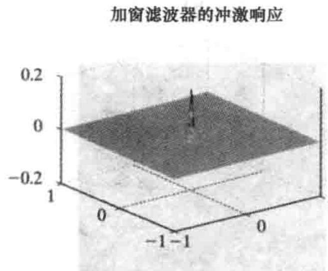
10.31



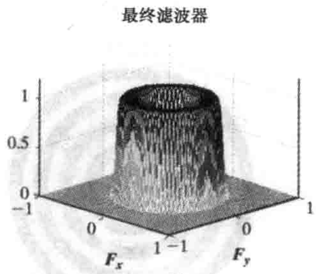
a）



b）



c）

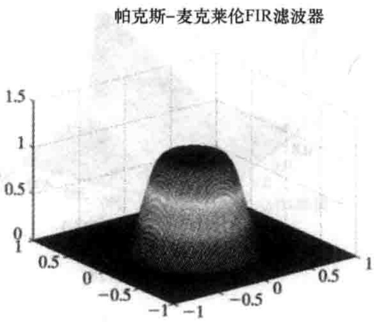


d）

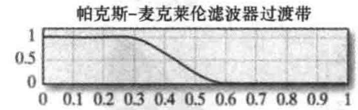
10.33 百分比波动＝12.7388％

10.35 对于频率抽样法，阶数 $N=46$。对于汉明窗，窗口宽度为 26，阶数为 25。

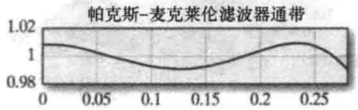
10.37



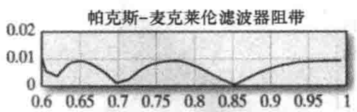
a）



b）

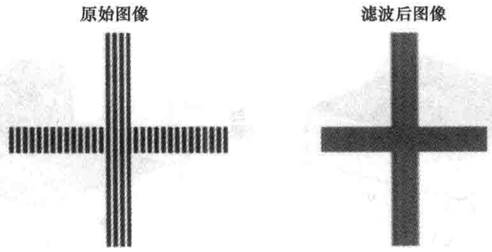


c）



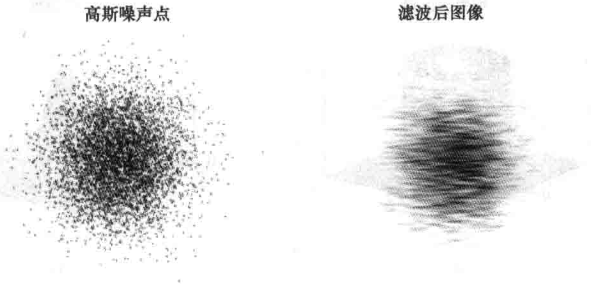
d）

10.39 滤波器阶数为 64。



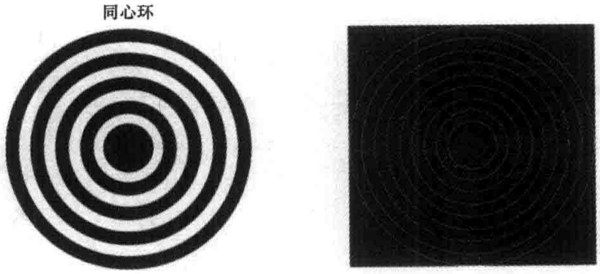
a) b)

10.41



a) b)

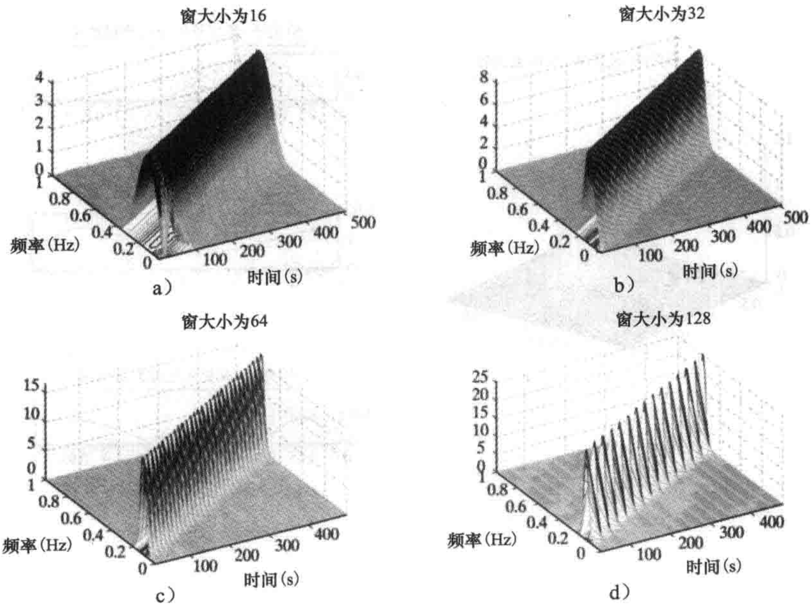
10.43



a) b)

第 11 章

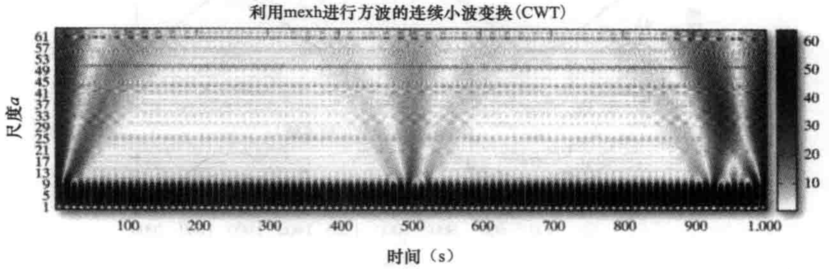
11.1



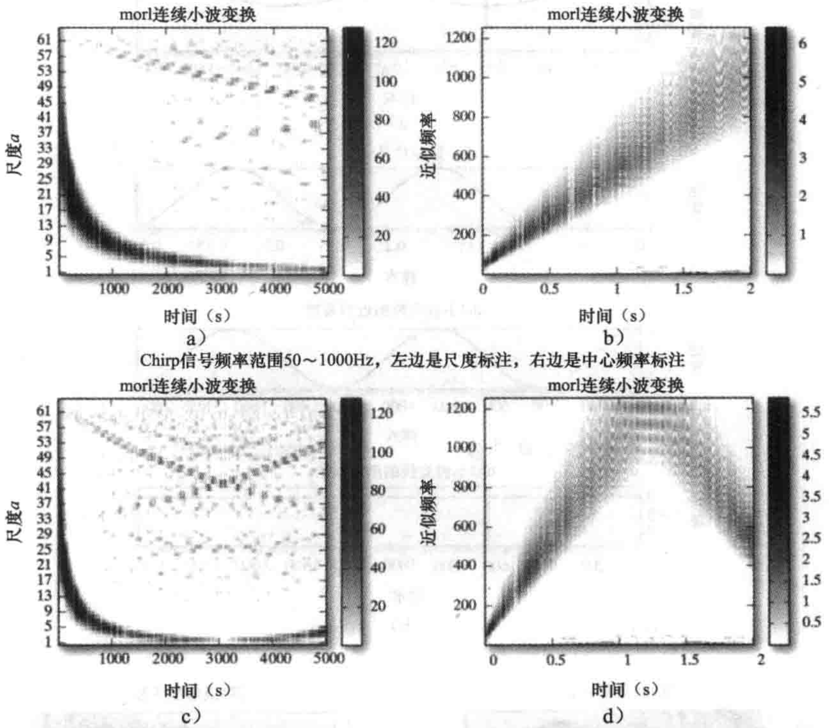
a) b) c) d)

11.3 自身的内积为 1，自身和位移信号的内积为 0。

11.5

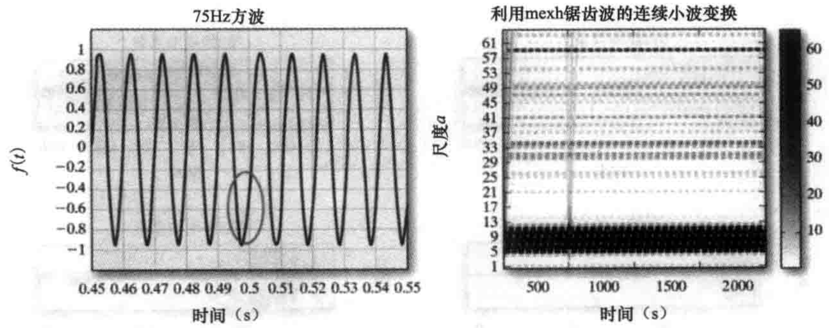


11.7



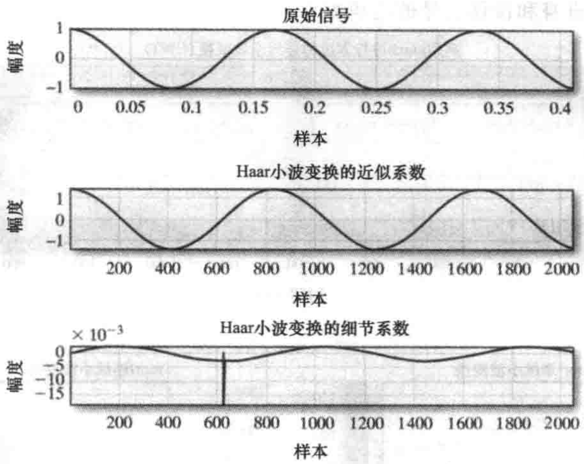
Chirp信号频率范围50～1000Hz，左边是尺度标注，右边是中心频率标注

11.9

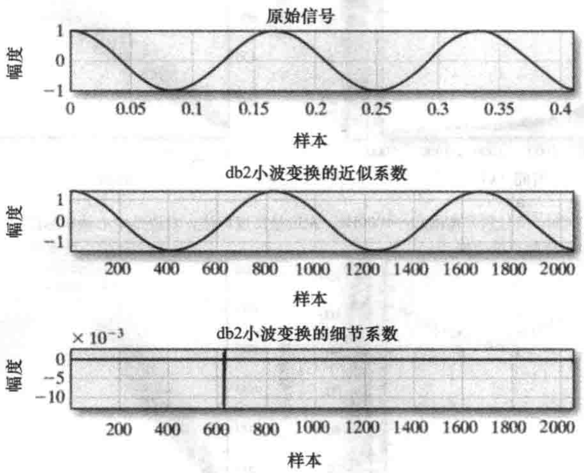


a） b）

11.11

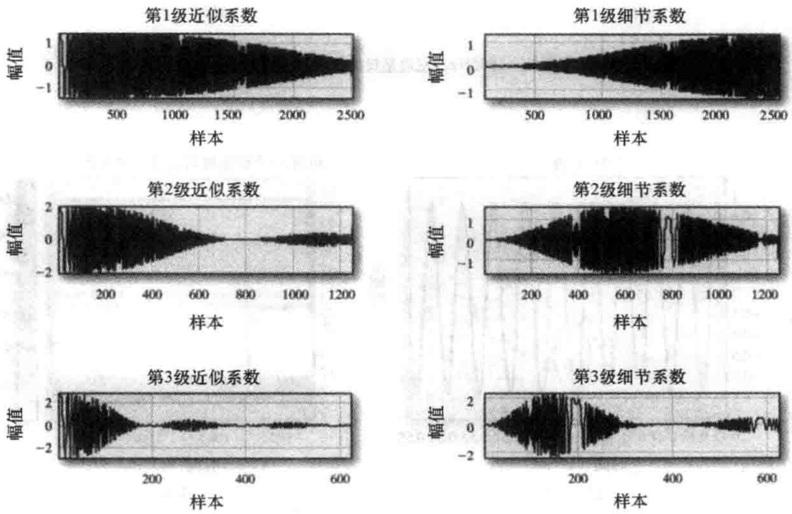

原始信号
幅度
样本
Haar小波变换的近似系数
幅度
样本
Haar小波变换的细节系数
× 10⁻³
幅度
样本

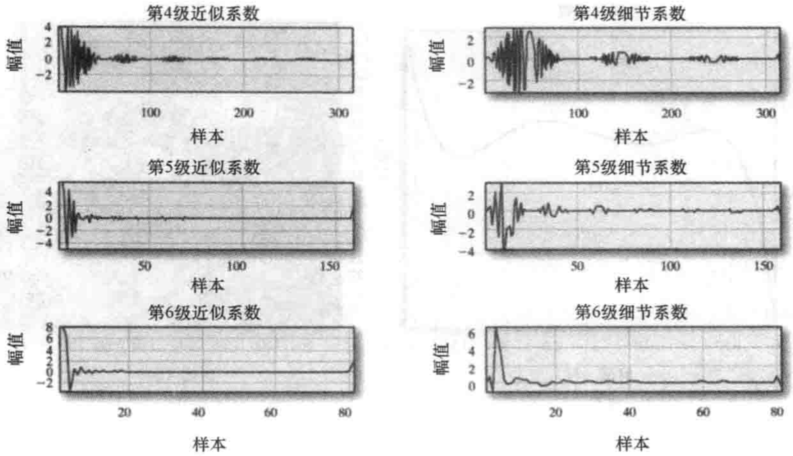
a)


原始信号
幅度
样本
db2小波变换的近似系数
幅度
样本
db2小波变换的细节系数
× 10⁻³
幅度
样本

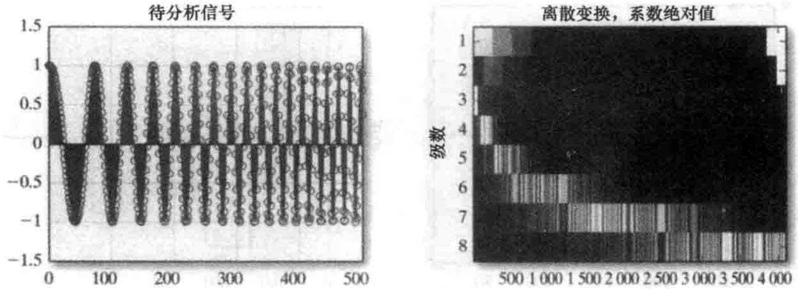
b)

11.13


第1级近似系数
第1级细节系数
第2级近似系数
第2级细节系数
第3级近似系数
第3级细节系数
幅值
样本

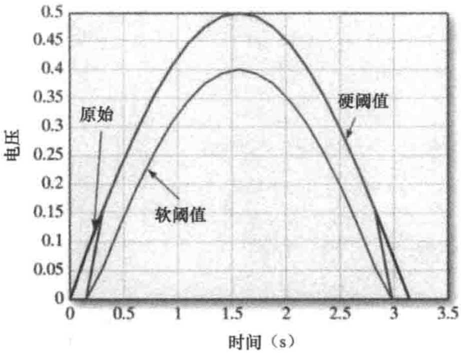


11.15



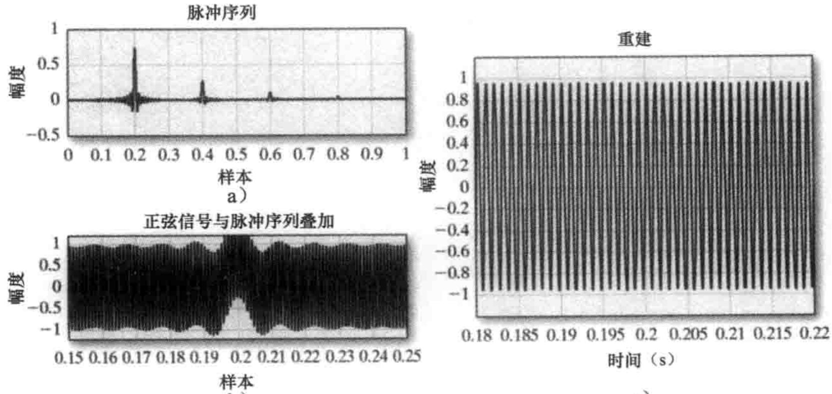
a） b）

11.17



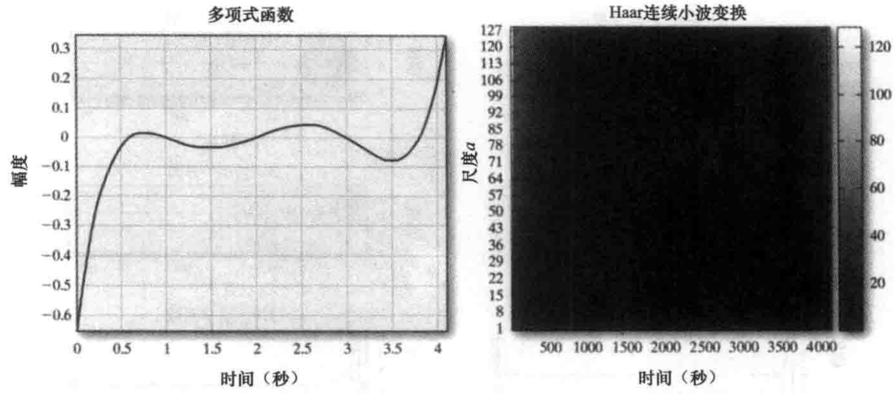
11.19 间断点都显现出来。Haar 小波函数使它们变得最明显。

11.21



a） b） c）

11.23



a） b）

11.25

